

3662 Vine Maple Dr. Eugene OR 97405

MARCH, 1986

Mike Dunn, Jim Bumpas, Larry Gold, co-editors

SMARTSHEET - Version 1.1 by Ken Shiu			
STATUS:	CELL:A1		
INPUT:	A	B	C
1			D
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

**MARCH MEETING
WED. 12TH
7:30PM
SOUTH EUGENE HIGH**

News and Reviews

by Mike Dunn, Co-Editor

Because our mailing list keeps growing, I again had to change to another program. The Atari Disk drives, even in double density are just not large enough to hold all your names. As you all know, each database program uses its own format on the files, with various delimiters to separate the fields and records. In the past, I've needed to write a program to do this, but lacking the time, I took a different approach this time. I decided to use dBase II on my CP/M ATR 8000, so I could use double or quad density double sided drives. Instead of using a null modem to port across the files, I used the excellent program from the ATR users group, The Capital Pro-Micro User's Group (Robert Danson, 2102 Bassett, Alexandria, VA 22308), called **ATOC**. You simply put your Atari disk in drive B, your CP/M disk in drive A, and copy disk to disk. Worked great. This program and many others are available from them. If you have an ATR 8000 with CP/M be sure to write them. Once the files were across, I converted them using WordStar copy and replace commands — not as easy as it sounds since it took many hours to figure out how to do it, but only a few minutes to do it. Now all is well — if any of you have the same problem, write me, and I'll tell you how to do it.

Speaking of writing, we have many User Group Newsletters giving various ways to add memory to your 800, 800XL and 130XE computers. If there is enough interest, we could put together a special newsletter with a disk of software to make it run, for say \$10. It will take a lot of work to do it, so please let me know if you want it.

Last weekend, I spent a few hours with my good friend Pat Warnshuis from the Portland Atari Group. Pat was the original editor of the Portland Atari Newsletter, and we have known each other since the "early days" of Atari. Anyway, he and a friend are developing an excellent program, a Wonder Editor for the ST, for programming in C and assembly, which has many marvelous features, including an outline ability and a 48 line display (on Monochrome). Before marketing it, he wants you to try it out, and let him know what you think. In addition, he wrote a memory test program which is especially useful if you have the new ROM or add memory to your ST. The disk can be obtained from Jim Bumpas for \$10; ask for "Pat's" disk.

For the 8-bit owners, Frank Pazel, the editor of the Jersey Atari Compter Group, has sent us the latest disk from his #1 programmers son. This double sided disk has useful Print shop utilities on one side, and on the back utilities which allow you to convert Print Shop files to Visualizer or MicroPanter. These disks are not full, but have AutoRun files on each side. \$10 for the disk.

Remember, all the ACE 8-bit disks are \$10 for 2 sides now.

Checkbook/Budget

Introduction: Non-profit organizations like churches have tax exempt status. Accounting is mainly a matter of keeping an accurate checkbook. Budgets are set up however so it is necessary to break payments into amounts designated for certain purposes. Income is also usually broken down in categories. This program was written to aid in keeping track of income and expenses in such situations and provide printouts of information when necessary. If you desire this much information on your household finances, you could easily set up the program for your personal use.

Hardware: This program should run on any Atari in the 400/800, XL or XE series but because of its length will require 48K. It is set up for use with a single disk drive. It is possible (with slight changes) to use a cassette drive but this is inconvenient because of the time required to load the program. A printer is not necessary because all information is available on the screen, but for normal use printed reports seem necessary.

Categories: Before using this program you must choose categories for income and expenses. Expenses have a category number associated with them. It is convenient to group similar items together (10's for salaries, 20's for buildings, 30's for supplies etc.). Once the expense categories are chosen, their names must be inserted in the program as data lines in lines 20000 - 20099. The last two digits of the line number are equivalent to the category number, ('20030 DATA Off. Supply' means office supplies have the category number 30). The maximum number of letters, including blanks, in a category name is 11.

Income is also broken up into categories representing the purpose for which certain amounts were given (Tithes, Missions etc.). In a personal account these could be different income sources: Husband's income, Wife's income, rebates, etc. There are no category numbers associated with income but the category names must be inserted consecutively as data lines in lines 20101 and following. The number of income categories is limited to seven because of a limited number of columns in the income summary. If you wish to use less than seven categories, change the variable NCAT in line 12012.

Check Numbers: Check numbers can range from 0001 through 9999. If you LIST the program you will find there are no line numbers in this range. The program creates data lines here using the associated check numbers. It then uses the 'LIST' command to save these lines and the 'ENTER' command to re-enter them. This makes the information easy to incorporate into other Basic programs. Deposit numbers can duplicate check numbers without problems. Deposit information is saved as data lines 10000 and greater. You will be asked to provide an associated check number with each deposit. This is the check number which contains the deposit entry in a normal check register.

Although the program will give you the balance after any check or a balance summary for the entire month, you should always keep a separate balance by hand to double check that everything got entered into the program correctly.

Reports: The program will print out any of four separate reports as well as a list of available categories. The reports are: Income Summary, Expense Summary, Category Totals and Balance Summary.

The income summary lists income categories in columns with each deposit as a separate row. Totals of rows and columns are provided. Examining income on the screen is a little different. You will be given a list of categories and asked to select one. Only information and a total for that particular category will be given.

The expenses summary is a list of each check including all pertinent information. If the check is to be applied to more than one category, the amount designated for each is given on a separate line. Output to the screen is similar but information is abbreviated because of space limitations. If you forget the numbers associated with various categories you can get a listing on the screen or printer.

A summary of category totals lists each category and gives amounts and check numbers of expenditures applied to that category. Category totals are also given. It is a cross-reference between amounts spent and check numbers. The same information is available on screen but in a different format. You are asked for a category number. Information for that category only is listed on the screen.

You can ask for the balance at any check number and be given the amount in the checking account before and after that check. You may continue with each check after that until the end of the month is reached. By starting with the first check of the month you can have a balance summary which is useful for checking against a balance done by hand.

Running the program: If the program does not automatically start, type 'RUN "D:CKBUDGET"' and hit RETURN. The program will ask you if you want to get information from disk or start from scratch with a clean slate. In either case, you will be asked to select a month. Information is kept on disk files by month having filenames 'D:CHECKS.XXX'; where XXX stands for the first three letter of the month name. A new 'D:CHECKS.FEB' will overwrite an old one so separate years should be kept on separate disks. You can add checks, modify checks, delete checks and do all sorts of printing of information; But unless you specifically call for the operation 'Write Month' the information on disk will not be updated! This should be the last thing you do before you turn the machine off or choose a 'New Month' operation. A 'New Month' selection starts the program over again, first going through and erasing all data lines having to do with checks and deposits.

Once you have selected a month, you will be asked to enter the beginning balance. If some information has already been entered for that month the beginning balance should already appear in parenthesis. In this case, just press 'RETURN' and the beginning balance will automatically be entered. This is normally the case on prompts for input. The value in parenthesis is the default response and is selected by pressing only 'RETURN'. Another convention involves choice selection from menus. The choices available will be indicated by letters (or key names, like RETURN), appearing in inverse video. 'RETURN' alone in response to a choice selection usually returns you to the main menu.

Main Menu: The main menu should appear after the initial balance has been entered. The following describe the various options:

Add Checks: You will be prompted for various check information including the amount. You are then to divide the check up into amounts in various categories. The program will ask for a category number and then an amount, repeating this sequence until the total check amount is accounted for. The default response for amount is always the remainder, so hitting only 'RETURN' will enter that amount and end the process. If you enter an amount which will make the total too high, you will be told and made to repeat the category breakdown from the beginning. If you can't remember a category number, you can enter a zero for the null category. It can always be fixed up later. The null category has no name associated with it.

BUMPAS REVIEWS

ZOOMRACKS

ZOOMRACKS (16-bit — Quickview Systems, 146 Main Street, Los Altos, CA 94022, \$80 list, but discounted to user groups until May 30 for as low as \$40 in quantities of 13 or more) is a novel type of data filer program. It makes use of a system never before seen on a computer as far as I know. This novelty is so smoothly executed, and so easy to use, I believe it is destined to become a milestone in personal computer history. It is useful enough to become a standard equivalent to Lotus 1-2-3 in the industry.

I don't believe I'll continue to use spreadsheets for a data manager any more. This program takes a concept from the factory floor, the time-card racks where employees store their time-cards near the time-card punch machine. Only here, each "card" can hold up to 250 lines of data. On a half-meg machine, each file can contain up to 9 racks, each containing over 400 cards. All racks can be displayed on the screen at once, but you might find this too much. There is a 10th rack, but this rack is always the directory of the disk. The data displayed on the screen is compacted so that even with 3 to 5 racks on the screen, "Thursday" might appear as "Thr".

When you click the mouse on an item, you can "zoom" up on it so it fills the screen with just one rack, or just one "Q" card in the rack ("Q" for "quick card"). Sorts are nearly instantaneous; as you enter a new record in any rack, it is automatically inserted in the sort order in the field you have selected. To resort is a simple click and two keystrokes away.

This is the first really and truly "free-form" data filer program I've seen. There are almost no rules to its use. Menus are available for the user not familiar with the commands. And the commands can be executed directly, without using the menus at all. When you first begin, you can also toggle on a "help" function which uses the bottom 4 lines of the screen to display information about the menu choices. There is also an excellent 128-page manual, and a long disk tutorial. The manual's table of contents is so complete, you probably won't miss the lack of an index.

I'm so excited about the imaginative concept of this program I'm afraid I can't fully communicate it. The screen doesn't just show you a title page and move you to a menu where you can "create" a database. The program starts up with a demo which places a screen full of racks and cards before you. There is a macro function in the program permitting you to use 2 keystrokes to enter repetitive data or commands. Or, you can create a custom application which automatically runs when you run the program.

When a rack is displayed, only the top line of data on each card appears. You can quickly re-assign which line is the "top" line. You can quickly pull up cards which relate to some data in another rack in the file. You can't really make a mistake more costly than a couple of keystrokes. Almost anything you do, you can undo with one keystroke. You can add, split and join fields, and change field names (labels) without losing any data. You can even change the name of a rack within a file.

The program can be used as a small word processor. The editing and formatting functions are very complete. You can produce mailing labels, you can create custom forms. You can use the mail merge function to print and address form letters. You have the option to save files in ASCII format which will permit the files to be used with other programs, or to transfer the data to other computers. The only addition to the power of the program I can imagine is to add some calculation function. Let's see now, how about a phone dialer to go with my list of addresses and phone numbers....

I highly recommend this program to anyone who needs to keep track of things — items, addresses, appointments, collections, things to do. You will think about ordering things in a totally different way than you've ever done before. You can be an anarchist, with very little external discipline to the way in which you handle your data and this program will bring order to your chaos. And the order will be created almost effortlessly.

Until May 30, 1986, **ZOOMRACKS** can be purchased by user groups at a discount for as low as \$40 each when the order includes 13 or more copies. Single copies can be bought for \$59 until March 31, or \$69 until April 30.

MARCH MEETING
WED. 12TH
7:30PM
SOUTH EUGENE HIGH

ANTETAM

ANTETAM (8-bit — SSI, \$50) is a dream come true for me. When I was a board war-gamer, my favorite system was SPI's "Terrible Swift Sword". This game system evolved through a number of successful and enjoyable cardboard games. Well, SSI has created a regimental-level Civil War simulation which contains all the elements which made those board games a success, and has added features not possible on cardboard.

Play options include the computer playing either (or both) sides. A two-player option is available, too. You can select one of 5 levels of difficulty, and 3 levels of play (Advanced level adds more features which complicate command control, and use of hidden units). This program also offers the option of using two disk drives. This is great! It minimizes disk swapping.

The cursor selects individual regiments and batteries. When selected, each unit has two "pages" of data. A "page" here is a 4-line window at the bottom of the screen. Current numbers of effective men in the regiment are listed along with morale and other status indicators such as facing, formation (column or line, mounted/dismounted, limbered/unlimbered), fortification level, ammunition, etc. The screen display shows unit facing in one of 4 positions. Actual facing can be in any one of 8 directions. So you have to read the status pages to actually see if a unit is facing "north" or "south".

One really great feature of this game (and which bedevils cardboard play) is maintaining command control. An option permits one to show all the regiments in one division with screen high-lighting. The regiment containing the commanding officer flashes! So it's an easy matter to determine if one is in command control. And you can see if your commander is too close to the front lines (or maybe hanging back too far in the rear — coffee, anyone?).

Infantry in line formation and dismounted cavalry are displayed as rectangles. Mounted cavalry and infantry in column are pairs of squares. Limbered artillery is a rectangle with a dot in the center. Unlimbered is a rectangular formation of 6 lines. Terrain includes hills, towns and roads, as well as woods, cornfields, rivers and streams. I was a little disoriented by the map at first, since the designers chose to put "South" at the top of the map. There is complete documentation, including Orders of Battle and a historical discussion. Also included is a laminated 4-color 8.5x11" conference map.

The game scale makes each turn an hour in the actual battle. You will probably be able to play one turn an hour. Think of it as a "real-time" game, at least in terms of player fatigue. Combat is executed sequentially, unit-by-unit, with the defender in each phase having the advantage of first fire. Normal execution times for each operation is 2 or 3 seconds. You can set a delay counter at any point from 1 to 9. I find 1 is too short to read the screen display of the results of actions. 2 seems about right after I became accustomed to the display. The default of 3 is best for new players, or for a more time to read the results.

This leads me to a serious problem I've experienced with the game in the solitaire mode (playing either the USA or CSA). I've not yet been able to play the game past Noon (game play begins with the 6:00 am turn). At this point, the program forgets it's using two disk drives and asks me to insert the other side of the disk. But more seriously, the delay in game operations increases by a factor of 10. Operations which formerly required only 2 or 3 seconds now require 20 to 30 seconds. This makes each turn require 2-3 hours to play. Obviously I only played one turn each game at this speed. I had to stop and start over. Even with normal play as intended, there are times you will have several minutes to do something away from the computer. Solitaire play requires letting the computer figure out its moves for 4-6 minutes, even with the screen display disabled.

Ordinarily I wouldn't review a game with this problem, as it doesn't seem ready for market in this condition. But SSI is acting so promptly with their concern over this problem, I'm satisfied they will correct the problem shortly (maybe by the time you read this). Perhaps it's my computer, since they did not discover this problem before release. If you have a similar problem, let SSI know about it and I am sure they will correct it.

— Jim Bumpas

ST ROMs

The ST ROMs arrived today (Feb. 13). I put mine in in less than a half an hour. How about 341k text buffer space with ST Writer? A little insert for your 520 ST user manual is included which covers some questions about LOGO. An instruction sheet explains how to install the ROMs. A disk is also included which contains system files for boot-up, along with the latest versions of NEO.PRG, BASIC.PRG, BASIC.RSC, SLIDENEOPRG, LOGO.PRG, and LOGO.RSC. The LOGO files seem to be the only ones with a different byte count than the files we already have.

ST Programmers

DataSoft (Marketing Project Manager, 19808 Nordhoff Pl., Chatsworth, CA 91311 818-886-5922) is advertising for programmers to submit "original graphic adventure games, and action arcade-style games full of color and animation. We also want practical home-productivity programs with mass-market appeal."

Once all necessary information is supplied you will be shown the information again in the form of a check and asked to confirm it. If you say 'yes' the information is correct, a data line will be entered containing that check's data. Any previous check of the same number will be replaced. Remember, the information has not been saved yet, only entered into the program. Once the data line is written you will be returned to the main menu and will have to select this option again to add another check.

Revise Checks: Use this option to make corrections or to delete checks. Enter the number of the check you want. Check information will be shown and you will be asked if it is correct. If it is, answer 'Y' for yes or just 'RETURN'. Answer 'N' (no) to make corrections. Answer 'E' to erase the check.

Deposits: Choose 'A' to add deposits made to the checking account. You will be asked for the total amount and then asked to specify how much to apply to each deposit category. The amount toward each category will be subtracted from the remaining amount until all is accounted for. 'R' will allow you to make corrections or Erase deposits previously entered and 'RETURN' will take you back to the main menu.

Income Summary: Choose 'I' to get a listing of the deposits broken down into amounts in various deposit categories. The listing can be to either the screen or printer. Screen listing is by individual category or a listing of the totals of all deposits. Before you select the printer, you may want to advance the paper to the top of the next page. If you do this you should inform the program by pressing 'ESC'. You can then press 'P' for printer.

Expenses Summary: 'E' will give a summary of all checks written for the month. You are again given the choice of screen, printer or 'ESC' to set the top of form.

New Month: Use this option if you wish to clear out all information entered and start again. I have on occasion found some data unerased by the program. It will probably be wise to do income and expense summaries to make sure everything was erased. Use the option again if anything remains.

Write Month: Be sure to use this option to save all information entered. Old information for the month will be overwritten.

Catagories: Four selections of category listings are available using the 'C' key. Individual totals of either deposit or expense categories is available on screen by choosing (C)heck category or (D)eposit category. If you want a printout of all expenses by category choose the (P)rint Category Totals option. A list of the categories themselves can be printed or sent to screen with (L)ist of categories option.

Balance Thru: You can use this option to find the balance before and after any check which has been entered. Values for the chosen check will be listed and you will be given a chance to continue (SPACE) or 'RETURN' to the main menu. The listing on screen can be toggled on and off with the SPACE key.

Technical: The program makes use of the Atari's ability to read information from its own screen. This allows it to modify itself by entering its own data lines. Data is entered in a data line corresponding to the check number. The data itself is a string with information as follows:

1-2, day; 3-22, payee; 23-30, amount; 31-48, memo; 49, number of subamounts; 50-51, 1st category; 52-59, 1st category amount; 60-119, remaining category numbers and amounts.

Deposit data is stored as data lines 10000 and following in steps of 10. The data is again a string as follows:

1-2, day; 3-6, deposit number; 7-14, amount; 15-18, associated check #; 19-26, 1st category amount; 27-74, remaining category amounts.

Another technique used frequently is the ability to restore to a particular data line. In certain cases it is necessary to remember what data line was being operated on. This information is contained in locations 183 and 184; the line number given by the quantity PEEK(183) + 256 * PEEK(184). Location 182 holds the index of data in a data line and is used to dig out the month in line 12270. Data is saved by 'LIST'ing the appropriate lines to disk. Data line 0 holds two numbers, the month # and the beginning balance for the month.

The maximum number of positions reserved for dollar amounts is eight. This limits any amount to a maximum of \$99,999.99.

— Stan Ockers

**MARCH MEETING
WED. 12TH
7:30PM
SOUTH EUGENE HIGH**

STuff

The really hot ST rumors this month revolve around the "Magic Box." An MS-DOS emulator for under \$200! It may contain up to 8 slots for various boards. One board will have an 8088 with 512k RAM. The Magic Box plugs into the DMA port. The rumors claim the screen might be slower than on an MS-DOS machine, but the higher processor speed should make up for this, causing the programs to actually run faster than on an original MS-DOS machine. I'll believe it when I see it.

The CP/M emulator is apparently ready. It should sell for about \$30. It's been appearing on various BBS around the country. But Sysops are removing it after discovering it's proprietary. Unfortunately, many of them are out there now. It's not enough to run CP/M though, is it? Don't you also need to buy the CP/M operating system? I've heard the emulator program requests you to insert the CP/M disk. Do you really want to use WordStar on your ST? Here's your chance.

A new 14" monitor is in the works, with both composite and RGB output. Use it for both your ST and your 130 XE.

FLASH : The March issue of BYTE magazine features the 1040 ST with front cover picture and articles. Check this one out as evidence of the resurgence of Atari on the micro market.

Atari Corp. has been down-playing ST sales figures, making it difficult for us to determine how many are sold. But there are hints. A DataQuest report cited in a recent issue of InfoWorld says 100,000 STs were sold in the first 90 days on the market. If this 90 days covers August, September and October, then sales in 1985 should be well over 200,000. In fact, since the ST was sold outside the States for several weeks before it appeared here, that "first 90 days" could well have begun earlier than August, 1985.

In the first week of October, 1985 official word from Atari was that they were selling STs "at the rate" of 50,000 per month. So, November and December adds another 100,000 (at least) to the DataQuest figure. In fact, STs sales accelerated quite dramatically over the Christmas buying season, so probably many more than 100,000 were sold in those two months. The January, 1986 issue of "Feedback" (Adelaide, Australia ACE) reports 40,000 STs were sold in Germany alone during October, November and December, 1985. So that figure confirms these other estimates.

VIP UPDATE

VIP Professional (16-bit — VIP Technologies, 132 Aero Camino, Santa Barbara, CA 93117, \$122 discounted) has now reached dealers with the third version since its release. The latest one arrived just as we went to press last issue. Some of the problems we had with the program are now corrected. The formula cells now calculate beyond the 80th character in the cell. Our calculations are no longer truncated. And the data buffer has been expanded to more than 40k. I am informed by a friend who has ROMs installed, that the data buffer on a ROM-based 520 ST is less than 80k!

I'm a little alarmed by this information. It make me wonder where our half-megabyte of memory went. If the OS is entirely (or mostly) on ROM, and the spreadsheet program only takes 277k on the disk, where is the other 200k RAM? The "Read Me First" pamphlet which comes in the VIP package suggests we'll have all but a small portion of the 512k available for applications when the ROMs are in. I want to know what happened? Here's how it's been explained to me. Start with 512k. VIP takes 277K program code and sets up two 32k graphics buffers for a total of 64k. This is why you can flip back and forth from the template to the graph screen so quickly. This leaves about 171k. The OS takes some space (more, depending upon how many desktop accessories you have), reducing you to between 80k and 108k for the spreadsheet.

I've discovered a couple of other problems with the VIP program since last issue. The Graph function from within the program works very smoothly and quickly in most of its options. But I can't get it to "Use" a previously created graph file which I saved to disk. I get a file not found error.

My data disk also contains 3 spreadsheet templates which I can no longer load in. When I try, the program says there's something wrong with my drive. I get the same message on either drive. When the error occurs I can't get out of it. Pressing Esc or Return gives me a system error 3, and then a system error 2. The only function which will work is the Help function. I have to reboot to get out of it. These data templates were created with a previous version of VIP. But other early templates on the same disk are loaded in with no problem. I'm perplexed.

WARNING

We received a warning from VIP today. They say their contract with Shanner, also doing business as "Gumball Express," is no longer legally permitted to sell the VIP Professional spreadsheet. If you buy from them, VIP says you'll be getting an old version, and VIP will not honor any warranty. To get your money back, you're asked to contact the California Attorney General's office in Sacramento, California.

SMARTSHEET BY KEN SHIU

```

1 REM #####
2 REM # SMARTSHEET VERSION 1.1    #
3 REM # by Ken Shiu      #
4 REM # Published by Atari Computer #
5 REM # Enthusiasts (N.S.W.)   #
6 REM # June 1985      #
7 REM #####
8 DIM OUT$(4800),IN$(4800),SS(600),PL
9 I$(20),U$(80),H$(15),H(4),CELL$(3),C$(8)
10 A$(11),F$(14),D$(20),PR$(120)
11 GRAPHICS 8:POKE 559,0:OPEN #2,4,0,"K:";POKE 752,1:POKE 718,127:POKE 789,0
12 :POKE 712,56:POKE 82,8
13 DL=PEEK(560)+PEEK(561)*256:POKE DL+7,130:DLIS="H$1600,A$1280,B$1280,C$1280,D$1280,E$1280,F$1280,G$1280,H$1280,I$1280,J$1280,K$1280,L$1280,M$1280,N$1280,O$1280,P$1280,Q$1280,R$1280,S$1280,T$1280,U$1280,V$1280,W$1280,X$1280,Y$1280,Z$1280":A=AD
14 R(DLIS):DIM CL$(8)
15 B=INT(A/256):C=A-B*256:POKE 512,C:P
16 OKE 513,B:POKE 54286,192:GOSUB 2000
17 30 ? "5 - SMARTSHEET - Version 1.1 by Ken Shiu STATUS?";:POKE 85,31:? "CELL"
18 :"CELL$"
19 35 ? " INPUT ":"? "
20 40 FOR T=Y TO Y+36 STEP 2:? " ";V$(T,T+1):NEXT T
21 45 IF A$=" " THEN ? "
22 46 IF FL THEN RETURN
23 50 TRAP 50:POKE 53774,64:POKE 16,64:IF PEEK(764)<>255 THEN 60
24 54 IF PEEK(53279)=5 THEN POKE 53760,10
25 56 IF PEEK(53279)=6 THEN POKE 53760,50
26 57 IF PEEK(53279)=3 THEN POKE 53760,15
27 58 GOTO 50
28 60 GET #2,K
29 65 IF K=29 THEN GOSUB 200:Y=Y+1:GOSUB 100
30 70 IF K=28 THEN GOSUB 200:Y=Y-1:GOSUB 100
31 75 IF K=30 THEN GOSUB 200:X=X-1:GOSUB 150
32 80 IF K=31 THEN GOSUB 200:X=X+1:GOSUB 150
33 85 IF (K)>47 AND K<(58) OR K=45 OR K=43 THEN GOSUB 900
34 90 IF (K)>64 AND K<(123) OR K=34 THEN GO SUB 220
35 92 IF K=58 THEN GOSUB 800
36 95 POKE 764,255:POKE 702,64:GOTO 50
37 100 IF Y>40 THEN Y=40:FOR T=1 TO 10:POKE T,128:NEXT T
38 KE 53761,168:NEXT T:POKE 53761,160:GOSUB 135:RETURN
39 105 IF Y<1 THEN Y=1:FOR T=1 TO 10:POKE (CP,CP+7)
40 110 CELL$=CHR$(ASC(H$(X,X))-128):CELL$=(LEN(CELL$)+1)=STR$(Y):POSITION 36,1:?" ";CELL$:Y1=Y
41 115 CP=(X-1)*8+(120*(Y-1))+1:IF Y1>20 AND K=29 THEN Y1=20
42 116 IF Y1=YMIN AND Y=YMAX THEN Y1=20
43 117 IF Y<YMIN THEN Y1=1
44 118 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
45 119 IN$+1:POSITION 0,4:?" ";POSITION 0,23:?" ";U$(Y*2-1,Y*2);OUT$(CP,CP+31);"?":RETURN
46 120 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
47 121 IN$+1:POSITION 0,4:?" ";POSITION 0,23:?" ";U$(Y*2-1,Y*2);OUT$(CP,CP+31);"?":RETURN
48 122 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
49 123 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
50 124 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
51 125 IF Y>YMIN THEN YMIN=YMIN-1:YMAX=YM
52 126 AX-1:POSITION 0,4:?" ";POSITION 0,4:?" ";U$(Y*2-1,Y*2);OUT$(CP,CP+31);"?":RETURN
53 127 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
54 128 IF Y>YMAX THEN YMAX=YMAX+1:YMIN=YM
55 129 IN$+1:POSITION 0,4:?" ";POSITION 0,23:?" ";U$(Y*2-1,Y*2);OUT$(CP,CP+31);"?":RETURN
56 130 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
57 131 POSITION 8,1:?" "
58 132 POSITION 8,2:?" "
59 133 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
60 134 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
61 135 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
62 136 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
63 137 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
64 138 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
65 139 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
66 140 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
67 141 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
68 142 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
69 143 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
70 144 IF A$="2" THEN ? "Value?";INS(CP,C$)=CHR$(K)
71 145 IF A$=" " THEN ? "
72 146 IF A$="3" THEN ? "Formula?";INS(CP,C$)=CHR$(K)
73 147 IF A$="4" THEN ? "
74 148 RETURN
75 149 IF K>27 AND K<32 THEN GOSUB 210:POP P:GOTO 65
76 150 IF X>15 THEN X=15:FOR T=1 TO 10:POKE T,128:NEXT T
77 151 IF X>15 THEN X=15:FOR T=1 TO 10:POKE T,128:NEXT T
78 152 IF X>15 THEN X=15:FOR T=1 TO 10:POKE T,128:NEXT T
79 153 IF X>15 THEN X=15:FOR T=1 TO 10:POKE T,128:NEXT T
80 154 IF X>15 THEN X=15:FOR T=1 TO 10:POKE T,128:NEXT T
81 155 IF X<1 THEN X=1:FOR T=1 TO 10:POKE T,128:NEXT T
82 156 IF X<1 THEN X=1:FOR T=1 TO 10:POKE T,128:NEXT T
83 157 IF X<1 THEN X=1:FOR T=1 TO 10:POKE T,128:NEXT T
84 158 IF X<1 THEN X=1:FOR T=1 TO 10:POKE T,128:NEXT T
85 159 IF X<1 THEN X=1:FOR T=1 TO 10:POKE T,128:NEXT T
86 160 CELL$=CHR$(ASC(H$(X,X))-128):CELL$=(LEN(CELL$)+1)=STR$(Y):POSITION 36,1:?" ";CELL$:X1=X
87 161 CP=(X-1)*8+(120*(Y-1))+1
88 162 IF X1=XMIN AND X1<=XMAX THEN X1=X
89 163 X=XMIN-1:GOTO 190
90 164 IF X1<XMIN THEN X1=1:XMIN=XMIN-1:X
91 165 MAX=XMAX-1
92 166 IF X1>XMAX THEN X1=4:XMAX=XMAX+1:X
93 167 IF X1>XMAX THEN X1=4:XMAX=XMAX+1:X
94 168 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
95 169 POSITION H(X1)-3,Y1+3:FOR T=CP TO CP+7:C$(T-CP+1,T-CP+1)=CHR$(ASC(OUT$(T,CP+7))):OUT$(CP,CP+7)=CL$(T,CP+7)+1
96 170 IF X1<XMIN THEN X1=1:XMIN=XMIN-1:X
97 171 IF X1>XMAX THEN X1=4:XMAX=XMAX+1:X
98 172 G=0:FOR T=XMAX-3 TO XMAX:G=G+1:POS ITION H(G),3:?" ";H$(T,T):NEXT T:G=0
99 173 G=0:T=(XMIN-1)*8+(120*(YMIN-1))+1
100 174 POSITION 4,G+3:?" OUT$(T,T+31);:NEXT T
101 175 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
102 176 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
103 177 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
104 178 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
105 179 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
106 180 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
107 181 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
108 182 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
109 183 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
110 184 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
111 185 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
112 186 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
113 187 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
114 188 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
115 189 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
116 190 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
117 191 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
118 192 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
119 193 ? C$:?POSITION 8,1:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
120 194 IF A$="2" THEN ? "Value?";INS(CP,C$)=CHR$(K)
121 195 ? C$:?POSITION 8,2:T=15*(Y-1)+X:A$=SS(T,T):IF A$="1" THEN ? "Label?";INS(CP+T)=CHR$(K):C$(T+1,T+1)=CHR$(K+128)
122 196 IF A$=" " THEN ? "
123 197 IF A$="3" THEN ? "Formula?";INS(CP,C$)=CHR$(K)
124 198 RETURN
125 199 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
126 200 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
127 201 POSITION 8,1:?" "
128 202 POSITION 8,2:?" "
129 203 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
130 204 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
131 205 POSITION 8,1:?" "
132 206 POSITION 8,2:?" "
133 207 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
134 208 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
135 209 POSITION 8,1:?" "
136 210 POSITION 8,2:?" "
137 211 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
138 212 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
139 213 POSITION 8,1:?" "
140 214 POSITION 8,2:?" "
141 215 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
142 216 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
143 217 POSITION 8,1:?" "
144 218 POSITION 8,2:?" "
145 219 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
146 220 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
147 221 POSITION 8,1:?" "
148 222 POSITION 8,2:?" "
149 223 IF K=34 THEN T=-1:GOTO 235
150 224 POSITION 8+T,2:?" "
151 225 POSITION 8+T,2:?" "
152 226 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
153 227 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
154 228 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
155 229 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
156 230 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
157 231 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
158 232 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
159 233 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
160 234 POSITION H(X1)-3,Y1+3:CP=(X-1)*8+(120*(Y-1))+1:OUT$(CP,CP+7);:RETURN
161 235 GET #2,K:IF K=155 THEN GOSUB 210:RETURN
162 236 IF K>27 AND K<32 THEN GOSUB 210:POP P:GOTO 65
163 237 T=T+1:IF T>126 THEN 295
164 238 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
165 239 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
166 240 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
167 241 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
168 242 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
169 243 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
170 244 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
171 245 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
172 246 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
173 247 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
174 248 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
175 249 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
176 250 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
177 251 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
178 252 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
179 253 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
180 254 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
181 255 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
182 256 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
183 257 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
184 258 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
185 259 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
186 260 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
187 261 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
188 262 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
189 263 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
190 264 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
191 265 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
192 266 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
193 267 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
194 268 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
195 269 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
196 270 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
197 271 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
198 272 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
199 273 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
200 274 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
201 275 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
202 276 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
203 277 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
204 278 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
205 279 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
206 280 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
207 281 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
208 282 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
209 283 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
210 284 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
211 285 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
212 286 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
213 287 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
214 288 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
215 289 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
216 290 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
217 291 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
218 292 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
219 293 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
220 294 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
221 295 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
222 296 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
223 297 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
224 298 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
225 299 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
226 300 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
227 301 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
228 302 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
229 303 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
230 304 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
231 305 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
232 306 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
233 307 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
234 308 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
235 309 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
236 310 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
237 311 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
238 312 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
239 313 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
240 314 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
241 315 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
242 316 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
243 317 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
244 318 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
245 319 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
246 320 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
247 321 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
248 322 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
249 323 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
250 324 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
251 325 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
252 326 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
253 327 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
254 328 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
255 329 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
256 330 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
257 331 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
258 332 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
259 333 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
260 334 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
261 335 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
262 336 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
263 337 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
264 338 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
265 339 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
266 340 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
267 341 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
268 342 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
269 343 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
270 344 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
271 345 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
272 346 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
273 347 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
274 348 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
275 349 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
276 350 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
277 351 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
278 352 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
279 353 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
280 354 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
281 355 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
282 356 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
283 357 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
284 358 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
285 359 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
286 360 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
287 361 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
288 362 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
289 363 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
290 364 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
291 365 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
292 366 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
293 367 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
294 368 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
295 369 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
296 370 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
297 371 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
298 372 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
299 373 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
300 374 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
301 375 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
302 376 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
303 377 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
304 378 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
305 379 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
306 380 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
307 381 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
308 382 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
309 383 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
310 384 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
311 385 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
312 386 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
313 387 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
314 388 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
315 389 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
316 390 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
317 391 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
318 392 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
319 393 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
320 394 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
321 395 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
322 396 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
323 397 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
324 398 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
325 399 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
326 400 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
327 401 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
328 402 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
329 403 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
330 404 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
331 405 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
332 406 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
333 407 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
334 408 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
335 409 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
336 410 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
337 411 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
338 412 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
339 413 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
340 414 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
341 415 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
342 416 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
343 417 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
344 418 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
345 419 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
346 420 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
347 421 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
348 422 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
349 423 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
350 424 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
351 425 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
352 426 T=T+1:IF T>126 AND T=8 THEN T=-1:C$=" "
353
```

SMART CON'T

```

"RN
 1 IF K>68 THEN ? " ";:GOTO 305
321 POSITION 13,1:? "Input Filename"
322 GET #2,K:IF K=32 THEN 340
325 IF K<65 OR K>98 THEN ? " ";:GOTO 3
22
327 T=1:GOTO 335
330 GET #2,K:IF K=126 THEN T=1:AS$=CL$:
POSITION 8,2:? AS$:GOTO 322
331 IF K=155 THEN 338
332 IF (K<48 OR K>90) OR (K>58 AND K<6
5) THEN ? " ";:GOTO 330
335 POSITION 8+(T-1),2:? CHR$(K):AS$(T,
T)=CHR$(K):T=T+1:IF T>8 THEN T=8
337 GOTO 330
338 FOR T=1 TO LEN(AS$):IF AS$(T,T)=" "
THEN G=LEN(AS$):AS$(T,:="";T=6
339 NEXT T:FI$="D":AS$(LEN(AS$)+1)="."$5
":FI$(LEN(FI$)+1)=AS$:GOSUB 205:GOSUB 2
10:CLOSE #1:RETURN
340 TRAP 2100:OPEN #1,6,0,"D:*..55":TRA
P 390
344 INPUT #1:DI$:POKE 54286,192:IF LEN
(DI$)<17 THEN 390
345 POSITION 8,2:? "FILE:":DI$(3,10)
346 GET #2,K:IF K=155 THEN AS$=DI$(3,10
):GOTO 338
347 IF K=27 THEN GOSUB 210:POP :RETURN
350 IF K=32 THEN 344
355 GOTO 346
 1 POSITION 8,2:? "NO MORE SMARTSHEET
 FILES":CLOSE #1:FOR T=1 TO 300:NEXT
T
395 GOSUB 210:GOSUB 205:TRAP 40000:POP
:RETURN
400 POSITION 8,1:? "Load:"::GOSUB 308
405 TRAP 2100:OPEN #1,4,128,FI$:TRAP 4
20:POSITION 9,1:? "Loading...":AS$:T=7
410 A$=ADR(IN$):L=4800:GOSUB 570:OUT$=I
N$:
415 A$=ADR(S$):L=600:GOSUB 570:CLOSE #6
/16:POKE 54286,192
420 POKE 559,0:GOSUB 205:TRAP 40000:G
SUB 2020:FL=1:GOSUB 30:FL=0:POKE 559,0
625 IF K=126 THEN POSITION 8,2:? " "
630 IF K=155 AND T>1 THEN 645
635 IF T=4 THEN ? " ";:GOTO 615
640 GOTO 610
645 IF VAL(AS$(2))>40 THEN ? " ";:GOSUB
210:GOTO 600
650 DI$=A$:GOSUB 210:POSITION 8,1:? "P
rinter Codes? (Y/N)"
655 GET #2,K:IF K=78 THEN 705
660 IF K>89 THEN ? " ";:GOTO 650
662 TRAP 662:GOSUB 210:POSITION 8,1?:
"Enter Line Number    ":POSITION 8,2:
665 IF L<1 OR L>40 THEN ? " ";:GOTO 662
667 AS$=PR$(L*3-2,L*3):POSITION 14,1?:
"Printer Codes"
458 A$=ADR(S$):L=600:GOSUB 570:CLOSE #6
/16:POKE 54286,192
460 GOSUB 205:TRAP 40000:RETURN
500 POSITION 8,1:? "Menu: G L S E P H
? ";
505 GET #2,K
510 IF K=71 THEN 260
512 IF K=76 THEN 480
514 IF K=83 THEN 450
516 IF K=69 THEN 550
518 IF K=80 THEN 600
520 IF K=27 THEN GOSUB 205:RETURN
522 IF K=72 THEN 580
524 IF K=63 OR K=47 THEN 1500
545 ? " ";:GOTO 505
550 POSITION 8,1:? "Erase: Sure? (Y/N)
";
555 GET #2,K
560 IF K=89 THEN POKE 559,0:GOSUB 2000
:POP :GOTO 30
565 GOSUB 205:RETURN
570 G=16:CB=832+G:POKE CB+2,T:HI=INT(A
/256):LO=A-HI*256:POKE CB+4,LO:POKE CB
+5,HI
572 HI=INT(L/256):LO=L-256*HI:POKE CB+
8,LO:POKE CB+9,HI:I=USR(ADR("hhh$31v$"))
,6)
575 RETURN
580 GOTO 420
600 POSITION 8,1:? "Print: Lower Corne
r?":POSITION 8,2:T=1:AS$=""
605 GET #2,K:IF K=27 THEN GOSUB 205:RE
TURN
607 IF K<65 OR K>79 THEN ? " ";:GOTO 6
05
610 ? CHR$(K):AS$(T,T)=CHR$(K):T=T+1
615 GET #2,K:IF K<48 OR K>57 AND K>12
6 AND K>155 THEN ? " ";:GOTO 615
620 IF T=2 AND K=48 THEN ? " ";:GOTO 6
15
625 IF K=126 THEN POSITION 8,2:? " "
630 IF K=155 AND T>1 THEN 645
635 IF T=4 THEN ? " ";:GOTO 615
640 GOTO 610
645 IF VAL(AS$(2))>40 THEN ? " ";:GOSUB
210:GOTO 600
650 DI$=A$:GOSUB 210:POSITION 8,1:? "P
rinter Codes? (Y/N)"
655 GET #2,K:IF K=78 THEN 705
660 IF K>89 THEN ? " ";:GOTO 650
662 TRAP 662:GOSUB 210:POSITION 8,1?:
"Enter Line Number    ":POSITION 8,2:
665 IF L<1 OR L>40 THEN ? " ";:GOTO 662
667 AS$=PR$(L*3-2,L*3):POSITION 14,1?:
"Printer Codes"
670 POSITION 8,2:? "Line ";L;" (E";AS$(1,1);":E";AS$(2,2);":E";AS$(3,3));": (44";
:T=1
675 GET #2,K:IF K=155 THEN 690
680 IF K=126 THEN 670
682 IF T>3 THEN ? " ";:GOTO 675
685 ? "E":CHR$(K):AS$(T,T)=CHR$(K):T=T
+1:GOTO 675
690 PR$(L*3-2,L*3)=AS$:GOSUB 210:POSITI
ON 14,1:? "More Codes? (Y/N)"
692 GET #2,K:IF K=89 THEN 662
695 IF K>78 THEN ? " ";:GOTO 692
700 TRAP 40000
705 G=VAL(DI$(2)):AS$=DI$(1,1):FOR T=1
TO 15:IF ASC(AS$(T,T))-128=ASC(AS$) THEN
L=T*8:T=15
710 NEXT T:GOSUB 210:GOSUB 205
720 TRAP 2100:OPEN #1,8,0,"P":POSITION
8,1:? "Press RETURN to print"
725 GET #2,K:IF K>155 THEN GOSUB 210:
GOSUB 205:RETURN
730 LI=1:FOR T=1 TO G:AS$=PR$(T*3-2,T*3
)
735 FOR Z=1 TO 3:IF AS$(Z,Z)=" " THEN N
EXT Z:GOTO 740
736 ? #1,AS$(Z,Z);:NEXT Z
740 ? #1,OUT$(LI,L):L=L+120:LI=LI+120:
NEXT T:CLOSE #1
750 GOSUB 205:GOSUB 210:POKE 54286,192
:RETURN
800 $$(15*(Y-1)+X,15*(Y-1)+X)="3":T=0:
CP=(X-1)*8+(120*(Y-1))+1:POSITION 8,1:
? "Sum: ( - )"
805 POSITION 8,2:? "From Cell: (44";
:INPUT AS$
810 IF LEN(AS$)>3 OR LEN(AS$)<2 THEN ? " ";
:GOTO 805
812 IF ASC(AS$(1,1))<65 OR ASC(AS$(1,1))
>79 THEN ? " ";:GOTO 805
814 IF VAL(AS$(2))<1 OR VAL(AS$(2))>40 T
HEN ? " ";:GOTO 805
818 POSITION 13,1:? AS$
820 POSITION 8,2:? "To Cell : (44";
:INPUT DI$
822 IF LEN(DI$)>3 OR LEN(DI$)<2 THEN ? " ";
:GOTO 820
824 IF ASC(DI$(1,1))<65 OR ASC(DI$(1,1))
>79 THEN ? " ";:GOTO 820
826 IF VAL(DI$(2))<1 OR VAL(DI$(2))>40
THEN ? " ";:GOTO 820
827 IF AS$(1,1)<>DI$(1,1) AND AS$(2)>DI
$(2) THEN POSITION 8,2:? "Cells must
be linear!":GOSUB 298:GOSUB 210:GOTO 8
00
828 POSITION 17,1:? DI$
830 FI$="";AS$(LEN(AS$)+1)="";AS$(LEN(A
$)+1)=DI$:FI$(LEN(FI$)+1)=AS$

```

SMART CON'T

```

835 IF LEN(FI$)<8 THEN FI$(LEN(FI$)+1) 1009 POSITION 16+T,1:? A$:K=LEN(A$)-1: 1216 TRAP 40000:IF ASC(A$(1,1))>64 AND
="" " A$=VAL(A$):IF G OR Z THEN IN$(CP+,CP+) 1218 A$=VAL(A$):GOTO 1230
840 OUT$(CP,CP+7)=FI$:IN$(CP,CP+7)=FI$+K=A$:T=T+K+1:RETURN 1220 TRAP 1225:F=ASC(A$(1,1))-64:G=VAL(
1010 IF 0=1 THEN A=A+L A$(2)):H=(F-1)*8+(120*(G-1))+1:A=VAL(
850 GOSUB 210:GOSUB 205:RETURN 1011 IF 0=2 THEN A=L-A IN$(H,H+7)):GOTO 1230
900 SS$(15*(Y-1)+X,15*(Y-1)+X)="2":T=0: 1012 IF 0=3 THEN A=A\ML 1225 TRAP 1226:A=VAL(OUT$(H,H+7)):GOTO
CP=(X-1)*8+(120*(Y-1))+1:POSITION 8,1: 1013 IF 0=4 THEN A=L/A 1230
? "Value:":IN$(CP,CP+7):C$="[" 1014 IF 0=5 THEN A=INT((L^A)+0.5) 1238
902 G=0:GOSUB 210:IN$(CP,CP+7)=CL$:OUT 1015 IN$(CP,CP+7)=CL$:IN$(CP,CP+7)=STR 1226 A=0
$$(CP,CP+7)=CL$:IF K=126 THEN T=-1:GOTO $$(A):K=155:GOSUB 942:GOSUB 210:SS$(15*(
910 Y-1)+X,15*(Y-1)+X)="2":RETURN A$=CL$:A$=IN$(CP+Z,CP+7):IF ASC(A$(1,1))>64 AND ASC(A$(1,1))<80 THEN 12
905 POSITION 8+T,2:? CHR$(K):C$(T+1,T+ 1020 GOSUB 210:POSITION 8,2:? "Enter C 34
1):CHR$(K+128):IN$(CP+T,CP+T)=CHR$(K): e11:";:INPUT A$:IF LEN(A$)<2 OR LEN(A$ 1232 L=VAL(A$):GOTO 1240
OUT$(CP+T,CP+T)=CHR$(K) )3 THEN 1025 1234 TRAP 1235:F=ASC(A$(1,1))-64:G=VAL(
910 POSITION H(X1)-3,Y1+3:? C$: 1021 IF ASC(A$(1,1))<65 OR ASC(A$(1,1) A$(2)):H=(F-1)*8+(120*(G-1))+1:L=VAL(
920 GET #2,K:T=T+1:IF K=155 OR (K)>27 A )>80 THEN 1025 IN$(H,H+7)):GOTO 1240
ND K(32) THEN GOSUB 210:GOTO 940 1022 IF VAL(A$(Z))<1 OR VAL(A$(Z))>40 1235 TRAP 1236:L=VAL(OUT$(H,H+7)):GOTO
922 IF T>7 THEN ? "[":;GOTO 920 THEM 1025 1240
926 IF K=126 THEN 900 1024 GOTO 1026 1236 L=0
928 IF (K)>47 AND K<(58) THEN 905 1025 ? "[":;GOTO 1020 1240 TRAP 40000:IF 0=42 THEN A=A\ML
930 IF K=46 AND G=1 THEN 935 1026 K=LEN(A$)-1:IN$(CP+,CP+T+K)=A$:P 1242 IF 0=43 THEN A=A+L
932 IF K=46 THEN G=1:GOTO 905 0SITION 16+T,1:? A$:T=T+K+1:RETURN 1244 IF 0=45 THEN A=A-L
935 ? "[":;GOTO 920 1030 SS$(15*(Y-1)+X,15*(Y-1)+X)="3":GOS 1246 IF 0=47 THEN A=A/L
940 IF K=155 AND NOT T THEN RETURN UB 210:POSITION 8,2:? "Operation:[ ]" 1247 IF 0=94 THEN A=INT((A\L)+0.5)
941 IF NOT T THEN POP :GOTO 65 [ ] 1248 IF GL=0 THEN A$=STR$(A):GOTO 1270
942 TRAP 955:A$=VAL(IN$(CP,CP+7)):IF GL 1031 GET #2,K:IF K=43 THEN 0=1:A$="+":; 1250 IF GL=2 THEN 1260
=0 THEN A$=STR$(A):GOTO 948 GOTO 1040 1252 A=INT(A+0.5):A$=STR$(A):L=8-LEN(A$)
943 IF GL=2 THEN 945 1032 IF K=45 THEN 0=2:A$="-":GOTO 1040 1254 :GOTO 1270
944 A=INT(A+0.5):A$=STR$(A):L=8-LEN(A$ 1033 IF K=42 THEN 0=3:A$="*":GOTO 1040 1260 IF A=INT(A) THEN A$=STR$(A):A$(LE
1):GOTO 948 1034 IF K=47 THEN 0=4:A$="/":GOTO 1040 1262 IF INT(A*10)=A*10 THEN A$=STR(
945 IF A=INT(A) THEN A$=STR$(A):A$(LEN A$(2)):H=(F-1)*8+.00":GOTO 948 1035 IF K=94 THEN 0=5:A$="^":GOTO 1040 1264 A=INT(100*A+0.5)/100:A$=STR$(A)
946 IF INT(A*10)=A*10 THEN A$=STR$(A): 1036 ? "[":;GOTO 1031 1270 L=8-LEN(A$):OUT$(CP,CP+L)=CL$:OUT$(CP+
A$(LEN(A$)+1)="0":GOTO 948 1040 Z=0:IN$(CP+,CP+T)=A$:POSITION 16 CP+L,CP+7)=A$:RETURN
947 A=INT(100*A+0.5)/100:A$=STR$(A) 1042 GET #2,K:IF K=67 THEN GOSUB 1020: 1400 SUM=0:A$=IN$(CP+1,CP+3):H=3:IF A$+
948 L=8-LEN(A$):OUT$(CP,CP+L)=CL$:OUT$(CP,CP +7):TRAP 40000 1044 IF K=78 THEN GOSUB 1008:GOTO 1050 1405 IF A=F THEN 0=A-1:FOR Z=L TO G:TR
950 IF K=155 THEN RETURN 1045 ? "[":;GOTO 1042 AP 1416:H=0*8+(120*(Z-1))+1:V=VAL(IN$(H,H+7)):GOTO 1420
952 POP :GOTO 65 1050 GOSUB 210:OUT$(CP,CP+7)=IN$(CP,CP 1418 IF L=G THEN 0=L:FOR Z=A-1 TO F:TR
955 ? "[":;K=126:GOTO 900 1051 +7):RETURN AP 1416:H=Z*8+(120*(0-1))+1:V=VAL(IN$(H,H+7)):GOTO 1420
960 FOR L=1 TO 10:POKE 53761,175:NEXT 1200 POSITION 8,1:? " Calculating... 1416 TRAP 1417:V=VAL(OUT$(H,H+7)):GOTO
L:POKE 53761,160:POKE 53760,200:RETURN 1417 V=0
1000 T=0:CP=(X-1)*8+(120*(Y-1))+1:POSI 1205 NEXT T:GOTO 420 1420 SUM=SUM+V:NEXT Z:A=SUM:GOSUB 1248
TIOM 8,1:? "Formula:":IN$(CP,CP+7):G=0 1210 CP=(T-1)*8+1:A$=CL$:IF IN$(CP,CP) =RETURN
? :G=Z=1 1202 FOR T=1 TO 600:IF SS$(T,T)="3" THE 1428
0:GOSUB 205:RETURN 1208 W GOSUB 1210 1417 V=0
1004 POSITION 8,2:? "1st No.:@ell or [ " 1205 NEXT T:GOTO 420
"umber":;GET #2,K:IF K=27 THEN GOSUB 21 1210 CP=(T-1)*8+1:A$=CL$:IF IN$(CP,CP) =RETURN
1212 FOR A=CP TO CP+3:L=A$=IN$(A,A)):M=-1:? :?
1005 IN$(CP,CP+7)=CL$:IF K=67 THEN G=1 1211 1500 ? "[":; " - SMARTSHEET HELP SCREE
:GOSUB 1020:GOTO 1030 1212 FOR A=CP TO CP+3:L=A$=IN$(A,A)):M=-1:? :?
1006 IF K>78 THEN ? "[":;GOTO 1004 1213 IF L=42 OR L=43 OR L=45 OR L=47 OR L=9 1505 ? " NORMAL CURSOR ARROW KEYS CON
1007 GOSUB 1008:L=A:GOTO 1030 1214 THEN Z=A-CP+1:A=CP+3:0=L:GOTO 1216 TROL THE":? " CURSOR {LONG BLACK BAR}
1008 TRAP 1008:GOSUB 210:POSITION 8,2: 1214 A$((A-CP+1,A-CP+1)=CHR$(L):NEXT A ."
? "Enter No.:":;INPUT A$:IF LEN(A$)>4

```

CHECKBOOK/BUDGET

BY STAN OCKERS

```

98 REM *****
11991 REM *      CHECKBOOK/BUDGET      *
11992 REM *      S. OCKERS 1/86       *
11993 REM *      KEEPS TRACK OF CHECKS *
11994 REM *      BROKEN DOWN INTO AS MANY *
11995 REM *      AS SEVEN CATEGORIES EACH *
11996 REM *****
11997 REM
11999 REM ***** INITIALIZATION *****
12000 DIM TEMP$(120), PAYEES$(20), MEMO$(18), MONTH$(9), RESP$(20), DS$(17): DS$="" : DS$=I": CKNBR=1: STBAL=0
12010 DIM SUBAMT$(8), SUBLEG$(8), KEY$(1), SEL$(1), CAT$(2), DOL$(8): OPEN #1, 4, 0, "K": POKE 16, 64: POKE 53774, 64
12012 DIM DEPN(15), ASCK(15), DAMT(15), SUDDEP(7): FOR J=1 TO 7: SUBDEP(J)=0: NEXT J: INCAT=7: PCNT=1
12014 POKE 789, 4: POKE 710, 10
12100 REM ***** WIPE OUT OLD DATA *****
12110 POKE 752, 0: POKE 712, 116: RESTORE "COUNT=0: POKE 82, 2
12120 ? "K": ? :?
12130 GOSUB 16210: IF CKNBR>10999 THEN 12180
12140 ? CKNBR: IF COUNT<12 THEN 12130
12150 RESTORE CKNBR: COUNT=0: ? :? :? "C": ONT": ? :? "CLEARING OUT OLD CHECKS": POSITION 0, 0: POKE 842, 13: STOP
12160 POKE 842, 12: GOTO 12120
12180 ? :? :? "CONT": ? :? "CLEANING OLD CHECKS": POSITION 0, 0: POKE 842, 13: STOP
12190 POKE 842, 12: POKE 752, 1
12200 REM ***** GET SOME DATA *****
12210 DFLG=0: POKE 712, 54: ? "K": POKE 82, 11: POSITION 12, 5: ? "CHECKBOOK/BUDGET"
12212 ? :? "Read from storage": ? :? "Start from Scratch"
12220 GET #1, KEY: KEY$=CHR$(KEY): IF KEY TO 12550
$="R" THEN DFLG=1: GOTO 12240
12230 IF KEY$<"S" THEN 12210
12240 RESTORE 12390: POKE 712, 250: ? "K": FOR J=1 TO 12: READ MONTH$: J; " "; MO *****
12250 MONTH$=1: ? :? "Enter Month # ("; M ITION 11, 2: ? MONTH$; " Checks": ? :? "-----"
ONTH; "J": ;? " | Add Checks | "
12260 NB=2: GOSUB 16300: IF RESP$<">"" TH EN MONTH=NBR: IF MONTH<1 OR MONTH>12 TH EN 12240
12270 RESTORE 12390: POKE 182, MONTH-1: R EAD MONTH$: IF DFLG=0 THEN 12410
12300 REM ***** DATA FROM STORAGE *****
12310 POKE 82, 2: TRAP 12350: RESP$="D CH ECKS": ? :? RESP$(10)=MONTH$(1, 3): RESP$(2, 2)=CHR$(58)
12320 ? "K": ? :? "ENTER "; CHR$(34); RES P$: CHR$(34): ? :? :? ? :? "CONT": ? :? "EADING FROM STORAGE"
12322 POSITION 0, 0: POKE 842, 13: STOP
12330 POKE 842, 12: ? "K": RESTORE 0: READ MONTH, STBAL: TRAP 40000
12340 RESTORE 12390: POKE 182, MONTH-1: R EAD MONTH$: GOTO 12410
12350 POKE 842, 12: ? "K": ? :? "CAN'T FIND DATA": FOR J=1 TO 300: NEXT J: TRAP 40000 : GOTO 12200
12390 DATA JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
12400 REM ***** START FRESH *****
12410 POKE 82, 12: ? "K": POSITION 12, 8: ? "Starting Balance": ? :? (""; STBAL; "J"); :? :? B=8: GOSUB 16300
12412 IF RESP$<">"" THEN STBAL=NBR
12420 ? "K": ? :? "0 DATA "; MONTH"; ", "; STBAL: ? :? :? "CONT": POSITION 0, 0: POKE 842, 13: STOP
12430 POKE 842, 12: ? "K": POSITION 10, 9: ? :? "INSPECTING CHECKS"
12500 REM ***** SET INIT VALUES OF CK #'S *****
12510 LASTREV=1: RESTORE 1: GOSUB 16210: IF CKNBR<10000 THEN LASTREV=CKNBR
12520 GOSUB 16210: IF CKNBR<10000 THEN LASTADD=CKNBR: GOTO 12520
12530 LASTADD=LASTADD+1: RESTORE 0: READ MONTH, STBAL: RESTORE 12390: POKE 182, MONTH-1: READ MONTH$
12540 RESTORE 10000: DEPLN=10000: GOSUB 16210: IF CKNBR<11000 THEN DEPLN=CKNBR: DREV=VAL(TEMP$(3, 6)): LASTDEP=DREV
12550 GOSUB 16210: IF CKNBR<11000 THEN DEPLN=CKNBR: LASTDEP=VAL(TEMP$(3, 6)): GOSUB 16300
12560 DEPLN=DEPLN+10: CKNBR=LASTREV: LASTDEP=LASTDEP+1
12570 REM ***** MENU *****
12580 POKE 712, 54: ? "K": POKE 82, 11: POS "" THEN GOSUB 13230: PAYEE$=RESP$ 12590 ? :? " | Revise Checks | ":? DS$: ? :? Deposits | :? DS$: ? :? Income Summary | :? DS$: ? :? Expenses Sum. | "
12600 ? :? DS$: ? :? " | New Month | ":? DS$: ? :? Write Month | :? DS$: ? :? Categories | :? DS$: ? :? Balance Thru | "
12610 ? :? "-----"
12620 REM ***** GET CHOICE *****
12630 GET #1, KEY: SEL$=CHR$(KEY): IF SEL $="A" THEN CKNBR=LASTADD: GOSUB 13010: LASTADD=CKNBR+1
12640 IF SEL$="I" THEN GOSUB 14910
12645 IF SEL$="E" THEN POKE 712, 102: GOSUB 13410
12650 IF SEL$="R" THEN POKE 712, 86: CKN BR=LASTREV: GOSUB 16810: GOSUB 16810: GOSUB 13140: LASTREV=CKNBR+1
12670 IF SEL$="N" THEN GOTO 12100
12680 IF SEL$="W" THEN POKE 712, 250: GOSUB 14010
12690 IF SEL$="C" THEN POKE 712, 134: GOSUB 12910
12700 IF SEL$="B" THEN GOSUB 14110
12710 IF SEL$="D" THEN POKE 712, 150: GOSUB 14510
12720 GOTO 12580
12900 REM ***** CATEGORY CHOICES *****
12910 ? "K": POSITION 11, 7: ? :? "Check Category": ? :? "Deposit Category": ? :? "List of Categories"
12920 ? :? "Print Category Totals": ? :? "RETURN": GET #1, KEY: KEY$=CHR$(KEY): IF KEY$="C" THEN GOSUB 13610: RETURN
12930 IF KEY$="D" THEN GOSUB 14810: RETURN
12940 IF KEY$="L" THEN GOSUB 13902: RETURN
12950 IF KEY$="P" THEN GOSUB 13710: RETURN
12960 IF KEY=155 THEN RETURN
12970 GOTO 12910
13000 REM ***** ENTER CHECK INFO *****
13010 POKE 712, 198: POKE 82, 5: ? "K": POSITION 5, 8: MB=4: ? "Enter Check # ("; CKN BR; ") "; GOSUB 16300
13012 IF RESP$<">"" THEN CKNBR=NBR
13020 NB=2: ? "Enter Day ("; DAY; ") "; GOSUB 16300: IF RESP$<">"" THEN DAY=NBR
13022 IF DAY<1 OR DAY>31 THEN 13020
13030 NB=20: ? "Enter Payee ("; PAYEE$; ") "; GOSUB 16300: IF RESP$<">
13040 NB=7: ? "Enter Amount ("; AMT; ") "

```

```

;:GOSUB 16300:IF RESP$<>" " THEN AMT=MB 13300 REM ***** DELETE A CHECK *
***** R
13050 MB=18:? "Enter Memo (";MEMO$;"")" 13310 ? "Delete this Item (Y/N)?":GET
?: " ",:GOSUB 16300:IF RESP$<>" " H1,KEY:KEY$=CHR$(KEY):IF KEY$<>"Y" THE
THEN GOSUB 13230:MEMO$=RESP$ N RETURN
13060 NUMSUB=0:SUBTOT=0 13320 POKE 709,196:? "K":? ?: CKNBR?:?
13070 NUMSUB=NUMSUB+1:? "Category (";S :? ?: "CONT":POSITION 0,0:POKE 842,13:
UBLEG(NUMSUB);") ";:NB=2:GOSUB 16300 STOP
13080 IF RESP$<>" " THEN SUBLEG(NUMSUB) 13330 POKE 842,12
=NBR 13340 ? "K":POSITION 16,10?: CKNBR;" "
13090 SUBAMT(NUMSUB)=AMT-SUBTOT:? "AMO ELETED":FOR J=1 TO 100:NEXT J:RETURN ;CAT$??
unt ("SUBAMT(NUMSUB);") "
13100 NB=7:GOSUB 16300:IF RESP$<>" " TH *** 13400 REM ***** EXPENSES SUMMARY *****
EN SUBAMT(NUMSUB)=NBR 13410 RESTORE 1:TOT=0:GOSUB 16668
13110 SUBTOT=SUBTOT+SUBAMT(NUMSUB):IF 13430 IF PFLAG=1 THEN LPRINT :LPRINT :
SUBTOT<AMT THEN 13070 LPRINT "EXPENSE SUMMARY FOR THE MONTH
13120 IF SUBTOT>AMT THEN ? "TOO MUCH!! OF ";MONTH$ 13440 IF PFLAG=1 THEN LPRINT :LPRINT "
!":FOR J=1 TO 500:NEXT J:GOTO 13060 CK #: DATE MEMO PAYEE
13125 GOSUB 16710 13450 POKE 82,2:? DATE PAYEE
13130 REM ***** CONFIRM IF CORRECT AMOUNT CATEGORY":LPRINT
***** 13460 GOSUB 16210:IF CKNBR>9999 THEN G
13140 ? "K":POKE 82,4:POSITION 4,4:? " 13442 IF PFLAG=1 THEN PCNT=PCNT+4:GOSU
":FOR J 8 16610 13458 POKE 82,2:? "CK # DATE PAYEE
=1 TO 9 13450 POKE 82,2:? "CK # DATE PAYEE
13142 ? "I 13460 GOSUB 16210:IF CKNBR>9999 THEN G
":NEXT J:?" 13460 GOSUB 16210:IF CKNBR>9999 THEN G
13144 POSITION 26,5:? " ";CKNBR:POSIT 13470 LSAV1=PEEK(183):LSAV2=PEEK(184):
ION 22,6:? "Date";MONTH;"/";DAY:POSI 13480 ? CKNBR;" ";MONTH;"/";TEMP$(1,2)
TION 6,8:? "Pay to";PAYEES 13480 ? CKNBR;" ";MONTH;"/";TEMP$(1,2)
13146 POSITION 18,10:? TEMP$(23,30);" 13480 ? CKNBR;" ";PAYEES(1,14);" ";TEMP$(50,51);"
Dollars":POSITION 6,12:? "MEMO";MEMO ;" ";PAYEES(1,14);" ";TEMP$(50,51);"
$:POKE 82,2 ;TEMP$(52,59):TOT=TOT+AMT
13148 POKE 201,19:POSITION 2,16:FOR J= 13581 IF PFLAG=1 THEN GOSUB 13598
1 TO NUMSUB:RESTORE 20000+SUBLEG(J):RE 13582 IF NUMSUB>1 THEN FOR J=2 TO NUMS
AD RESP$:POS=42+10*j UB:GOSUB 13592:COUNT=COUNT+1:NEXT J
13150 ? RESP$;" ";TEMP$(POS,POS+7),:ME 13583 POKE 183,LSAV1:POKE 184,LSAV2:IF
KT J COUNT=19 THEN COUNT=0:IF PFLAG=0 THE
13160 POSITION 7,21:? "CORRECT? (Y/N) M GOSUB 16110:GOTO 13450
base/RETURN":GET #1,KEY:KEY$=CHR$(KEY 13585 GOTO 13460
):IF KEY$="Y" THEN 13200 13586 RESP$=STR$(TOT):GOSUB 16510:IF P
13170 IF KEY=155 THEN RETURN FLAG=1 THEN LPRINT :LPRINT "
13172 IF KEY$="E" THEN GOSUB 13310:RET Total Expenses ";"DOL
URN 13587 IF PFLAG=1 THEN PCNT=PCNT+1:GOSU
13180 IF KEY$="W" THEN 13010 B 16610
13182 GOTO 13160 13588 ? :" " Total Expenses ";"DOL
13190 REM ***** ENTER CHECK AS DATA $ 13589 IF PFLAG=0 THEN GOSUB 16110
13200 POKE 82,0:? "K":? ?: CKNBR;"DATA 13590 RETURN
";TEMP$? "CONT":POSITION 0,0:POKE 84 13592 RESTORE SUBLEG(J)+20000:READ RES
2,13:STOP P$:POS=40+10*j;? "
13210 POKE 842,12:RETURN ;TEMP$(POS,POS+1);" ";
13220 REM ***** CLEAN OUT COMMA 13594 ? TEMP$(POS+2,POS+9)
5 ***** 13596 IF PFLAG=1 THEN LPRINT "
13230 FOR J=1 TO LEN(RESP$):IF RESP$(J,
,J)="," THEN RESP$(J,J)=" " ;TEMP$(POS+2,POS+9);" ";RESP$ 13597 GOSUB 16610:RETURN
13240 NEXT J:RETURN

```

CHECK CON'T

```

RINT,:NEXT L
1410 RESP$=STR$(CATOT):GOSUB 16510:LP
RINT CAT;" TOT=";DOL$:GOSUB 16610
13820 POKE 183,LSAV1:POKE 184,LSAV2:PO
KE 182,NSAV:GOTO 13718
13900 REM ***** LIST OF AVAILABLE CATE
GORIES *****
13902 GOSUB 16660:IF RFLAG=1 THEN RETU
RN
13910 ? "CATEGORYS":RESTORE 20001:CO
UNT=0:IF PFLAG=1 THEN LPRINT :LPRINT :
LPRINT "CATEGORYS":LPRINT
13912 IF PFLAG=1 THEN PCNT=PCNT+3:GOSU
B 16610
13920 READ RESP$:IF RESP$(1,3)="***" T
HEN 13940
13921 CAT=PEEK(183)+256*PEEK(184)-2000
0:? CAT;" ";RESP$:COUNT=COUNT+1
13922 IF COUNT=21 THEN COUNT=0:IF PFLA
G=0 THEN GOSUB 16110?: "5"
13924 IF PFLAG=1 THEN LPRINT CAT;" ";
RESP$:GOSUB 16610
13930 GOTO 13920
13940 IF PFLAG=0 THEN GOSUB 16110:RETU
RN
13942 RETURN
14000 REM ***** SAVE DATA TO STORA
GE *****
14010 ? "K":RESP$="D CHECKS.":RESP$(10
)=MONTH$(1,3):RESP$(2,2)=CHR$(58):POSI
T "W 10,10:? "Saving ";RESP$
20 LIST RESP$,0,10999:RETURN
14100 REM ***** CALCULATE BALANC
ES *****
14110 ? "K":POKE 82,7:POSITION 7,7:? "
Balance Calculation":? :RESTORE 1:GOSU
B 16210:STCK=CKNBR
14112 DEPCNT=0:RESTORE 10000:FOR J=0 T
0 15:DAMT(J)=0:ASCK(J)=9999:NEXT J
14114 GOSUB 16210:IF CKNBR>10999 THEN
14120
14115 IF TEMP$(3,6)="9999" THEN 14114
14116 DEPN(DEPCNT)=VAL(TEMP$(3,6)):ASC
K(DEPCNT)=VAL(TEMP$(15,18)):DAMT(DEPCN
T)=VAL(TEMP$(7,14)):DEPCNT=DEPCNT+1
14118 GOTO 14114
14120 ? "Start check":STCK;" ";:NB=4
:GOSUB 16300:IF RESP$()"" THEN STCK=NB
R
14122 RESTORE STCK:GOSUB 16210:IF CKNB
R>STCK THEN ? :? " NOT FOUND! "? :60
TO 14120
14130 BBAL=STBAL:RESTORE 1:GOSUB 16660
:IF RFLAG=1 THEN RETURN
14140 FOR J=0 TO 15:IF ASCK(J)<STCK TH
EN BBAL=BBAL+DAMT(J)
14142 NEXT J
14150 GOSUB 16210:IF CKNBR>9998 THEM 1 4660
14160
14152 IF CKNBR<STCK THEN BBAL=BBAL-VAL
(TEMP$(23,30)):GOTO 14150
14160 POKE 82,8:POKE 201,10:? :"PREV
. BAL CKN AMOUNT NEW BAL":?
14162 IF PFLAG=1 THEN LPRINT :LPRINT :
LPRINT "PREV. BAL CKN AMOUNT
NEW BAL":LPRINT :PCNT=PCNT+3
14164 IF PFLAG=1 THEN GOSUB 16610
14170 GOSUB 14300:? :"SPACE to conti
nue or RETURN":GET #1,KEY:KEY$=CHR$(KE
Y):IF KEY=155 THEN RETURN
14200 POKE 764,255:GOSUB 16210:IF CKNB
R>9998 THEN GOSUB 16110:RETURN
14210 GOSUB 14300:IF PFLAG=1 THEN 1420
0
14212 IF PEEK(764)=255 THEN 14200
14220 POKE 764,255
14230 IF PEEK(764)=255 THEN 14230
14240 GOTO 14200
14300 FOR K=0 TO 15:IF ASCK(K)=CKNBR T
HEN GOSUB 14350
14310 NEXT K
14330 CKAMT=VAL(TEMP$(23,30)):? BBAL,C
KNBR,CKAMT:,IF PFLAG=1 THEN LPRINT BBA
L,CKNBR,CKAMT,
14332 BBAL=BBAL-CKAMT?: BBAL:IF PFLAG=1
1 THEN LPRINT BBAL:GOSUB 16610
14340 RETURN
14350 ? BBAL,"DEP",DAMT(K),:IF PFLAG=1
THEN LPRINT BBAL,"DEP",DAMT(K),
14360 BBAL=BBAL+DAMT(K):? BBAL:IF PFLA
G=1 THEN LPRINT BBAL:GOSUB 16610
14370 RETURN
14500 REM ***** ENTER A DEPOSIT * "
***** THEN 14610
14510 ? "K":POKE 82,14:POSITION 17,7:? "
DEPOSIT":? :? "Review/Change":? :? "
Add Deposit":? :? "RETURN"::DPNBR=DREV
14520 GET #1,KEY:KEY$=CHR$(KEY):IF KEY
$="A" THEN DPNBR=LASTDEP:GOSUB 14610:L
ASTDEP=LASTDEP+1:DEPLN=DEPLN+10:RETURN
14522 IF KEY=155 THEN RETURN
14530 RESTORE 10000:POKE 82,7:? :? "En
ter Deposit":("DPNBR");? :NB=4:GOSU
B 16300:IF RESP$()"" THEN DPNBR=NBR
14540 GOSUB 16210:IF CKNBR>10999 THEN
? :? :? "CONT":POSITION 0,0:POKE 842,1
? :? "Not Found":GOSUB 16110:RETURN
14550 IF VAL(TEMP$(3,6))>DPNBR THEN 1
14710 POKE 842,12:RETURN
14560 DAY=VAL(TEMP$(1,2)):DPAMT=VAL(TE
MP$(7,14)):ASCHK=VAL(TEMP$(15,18)):SAV
LN=DEPLN:DEPLN=CKNBR
14562 FOR J=1 TO 7:POS=11+8*XJ:SUBDEP(J
)=VAL(TEMP$(POS,POS+7)):NEXT J:GOSUB 1
14800 REM ***** DEPOSIT CATEGORY TOT
AL *****
14810 TOT=0:RESTORE 20101:? "K":? :?
FOR J=1 TO INCAT:READ RESP$:? J,RESP$:
14820 ? :? "Press category number.":GE

```

CHECK CON'T

```

T #1,DCAT:DCAT=DCAT-48:IF DCAT<1 OR DC 14);"    "TEMP$(15,18):GOSUB 16610      16548 L=LEN(RESP$):DOL$(9-L)=RESP$:IF
AT>INCAT+1 THEN 14810          15070 GOTO 15030          L<8 THEN FOR K=1 TO 8-L:DOL$(K,K)="
14828 IF DCAT=INCAT+1 THEN POS=7:?;"K" 15090 RESP$=STR$(TOT):GOSUB 16510:IF P 16550 RETURN
?:? :"DATE DEP # AMOUNT":? :RESTORE 1 FLAG=1 THEN LPRINT :LPRINT "    Total D
0000:GOTO 14840          15091 PCNT=PCNT+1:GOSUB 16610          16600 REM ***** FORMFEED SUBROUTINE *
14830 RESTORE 20100+DCAT:READ RESP$?:? 15092 ? :?"    Total Deposit = ";DOL$: 16610 PCNT=PCNT+1:IF PCNT>57 THEN FOR
"E 10000:POS=11+8*DCAT          IF PFLAG=0 THEN GOSUB 16110          L=1 TO 67:PCNT:LPRINT :NEXT L:PCNT=1
14840 GOSUB 16210:IF CKNBR>11000 THEN 15094 RETURN          16620 RETURN
?:RESP$=STR$(TOT):GOSUB 16510:? " TO 16000 REM ***** WHICH CHECK? ***          16650 REM ***** PRINTER? SUBROUTINE *
TAL = ";"DOL$:GOSUB 16110:RETURN          160000 REM ***** WHICH CHECK? ***
14850 AMT=VAL(TEMP$(POS,POS+7)):TEMP 16010 ? "K":POSITION 6,10:? "Input Che 16660 PFLAG=0:RFLAG=0:? "K":POSITION 4
$(1,2);";TEMP$(3,6);";TEMP$(POS, 16210 READ TEMP$:COUNT=COUNT+1:CKNBR=P 16682 IF KEY=27 THEN PCNT=1:POSITION 1
POS+7):TOT=TOT+AMT          EEK(183)+256*PEEK(184):RETURN          0,10:? "TOP OF FORM SET":FOR J=1 TO 10
14860 GOTO 14840          16290 REM ***** LIMITED STRING INPUT          16685 IF KEY$()>"5" THEN 16660
14900 REM ***** INCOME SUMMARY ** 14910 POKE 201,10:GOSUB 16660:IF RFLAG 16296 REM ***** IF NUMBER, RETURNED IN RE 16690 RETURN
*****          16110 POKE 82,7:? ?:?"Press SPACE to c 16700 REM ***** ENCODE TEMP$ WITH CK
=1 THEN RETURN          16200 REM ***** READ A DATALINE SUBRO 16685 DATA *****
14912 IF PFLAG=0 THEN GOSUB 14810:RETU 16298 UTIME *****          16710 TEMP$(1)=" ";TEMP$(120)=" ";TEMP
RN          16299 REM *****          16720 TEMP$(23)=DOL$:TEMP$(31)=MEMO$:TEMP$(49)=STR$(NUMSUB):FOR J=1 TO NUMSU
14920 LPRINT :LPRINT "INCOME S 14928 FOR J=1 TO 7:SUBAMT(J)=0:NEXT J: 16292 REM ***** MB IS # BLANK SPACES A
UMMARY FOR THE MONTH OF ";MONTH$:LPRIN 16294 REM ***** INPUT RETURNED IN RE 16730 TEMP$(POS)=STR$(SUBLEG(J)):RESP$=
T :PCNT=PCNT+3:GOSUB 16610          16295 LLOWED *****          16730 TEMP$(SUBAMT(J)):GOSUB 16510:TEMP$(
14930 FOR J=1 TO 7:SUBAMT(J)=0:NEXT J: 16296 REM ***** IF NUMBER, RETURNED IN B:POS=0+10*K
ROUTINE *****          16297 GET #1,KEY:IF KEY=155 THEN TEMP$          16740 TEMP$(2)=DOL$:NEXT J:RETURN
TOT=0:RESTORE 20101:FOR J=1 TO INCAT:R 16298 GET #1,KEY:IF KEY=155 THEN TEMP$          16750 REM ***** GET CHECK DATA *****
EAD RESP$:LPRINT RESP$,:NEXT J          16299 GET #1,KEY:IF KEY=155 THEN TEMP$          16760 TRAP 16890:RESTORE CKNBR:READ TE
14940 LPRINT " Total":RESTORE 10000:L 16300 POKE 752,8:RESP$="";FOR L=1 TO N 16770 MP$:GOSUB 17010:RESP$=PAYEE$:GOSUB 169
PRINT :PCNT=PCNT+1:GOSUB 16610          16301 B:? "_" ;NEXT L:FOR L=1 TO NB:? CHR$(3 10:PAYEE$=RESP$
14950 GOSUB 16210:IF CKNBR>11000 THEN 16302 J+VAL(TEMP$(POS,POS+7));NEXT J 16780 RESP$=MEMO$:GOSUB 16910:MEMO$=RE
14988          16303 GET #1,KEY:IF KEY=155 THEN TEMP$          16790 SP$=CHR$(183)+256*PEEK(184)
14960 FOR J=1 TO INCAT:POS=11+8*j:LPRI 16304 POKE 752,8:RESP$="";FOR L=1 TO N 16800 TRAP 40000:RETURN
NT TEMP$(POS,POS+7);:SUBAMT(J)=SUBAMT 16305 B:? "_" ;NEXT L:FOR L=1 TO NB:? CHR$(3 16810 16900 REM *** STRIP RESP$ OF TRAILING
J:J+VAL(TEMP$(POS,POS+7));NEXT J 16306 J+1:NFLG=0          16810 L=LEN(RESP$)
16790 LPRINT TEMP$(7,14):TOT=TOT+VAL(T 16307 GET #1,KEY:IF KEY=155 THEN TEMP$          16812 RESP$=MEMO$:GOSUB 16910:MEMO$=RE
EMP$(7,14)):GOSUB 16610:GOTO 14950 16308 =RESP$:GOTO 16360          16814 SP$=CHR$(183)+256*PEEK(184)
16798 FOR J=1 TO INCAT+1:LPRINT "____ 16312 IF KEY>47 AND KEY<58 AND POS=1 T 16816 CKNBR=PEEK(183)+256*PEEK(184)
HEM NFLG=1          16313 IF KEY=126 AND POS=2 THEN ? CHR$(30) ;"_" ;CHR$(30) ;:POS=1:RESP$="" 16818 16900 DAY=VAL(TEMP$(1,2)):PAYEE$=TEMP$(1
16800 R$(SUBAMT(J))          16314 IF KEY=126 AND POS>2 THEN ? CHR$(30) ;"_" ;CHR$(30) ;:POS=POS-1:RESP$=RES
16990 GOSUB 16510:LPRINT DOL$:NEXT J: 16320 IF KEY=126 AND POS>2 THEN ? CHR$(30) ;"_" ;CHR$(30) ;:POS=POS-1:RESP$=RES 16819 16920 IF RESP$(L,L)=" "
16992 RESP$=STR$(TOT):GOSUB 16510:LPRINT DOL 16330 IF POS=NB+1 THEN 16310          16920 IF RESP$(L,L)=" "
16994 $:PCNT=PCNT+1:GOSUB 16610          16331 IF KEY=126 AND POS=2 THEN ? CHR$(30) ;"_" ;CHR$(30) ;:POS=1:RESP$="" 16920 IF RESP$(L,L)=" "
15000 REM ***** PRINT DEPOSIT SUM 16332 IF KEY=126 AND POS>2 THEN ? CHR$(30) ;"_" ;CHR$(30) ;:POS=POS-1:RESP$=RES 16920 IF RESP$(L,L)=" "
MARY *****          16333 IF POS=NB+1 THEN 16310          16922 P$(1,L-1):L=L-1:GOTO 16920
15010 TOT=0:? :PRINT "Dep # Date AMO 16340 IF KEY<20 OR KEY>122 THEN 16310          16930 RETURN
16930 UN ASSOC. CKNR":RESTORE 10000:PRINT 16350 ? CHR$(KEY):RESP$(POS,POS)=CHR$( 16940 17000 REM ***** DECRYPT CK DATA FROM
15020 IF PFLAG=1 THEN LPRINT :LPRINT : 16351 :KEY):POS=POS+1:GOTO 16310          16942 TEMP$ *****
15021 (KEY):POS=POS+1:GOTO 16310          16360 MBR=0:IF NFLG=1 THEN MBR=VAL(TEM 16944 L=LEN(RESP$)
15022 LPRINT "Dep # Date Amount ASSOC. CKNR":RESTORE 10000:PRINT 16361 P$ 16946 16950 IF RESP$(L,L)=" "
15030 16362 P$ 16948 IF RESP$(L,L)=" "
15030 GOSUB 16210:IF CKNBR>11000 THEN 16370 ? :POKE 752,1:RETURN          16950 IF RESP$(L,L)=" "
15031 16371 16952 IF RESP$(L,L)=" "
15040 DPNBR=VAL(TEMP$(3,6))          16500 REM ***** PUT RESP$ IN DOLLAR F 16954 P$(1,L-1):L=L-1:GOTO 16950
15041 16501 DPNBR=VAL(TEMP$(3,6))          16502 ORMAT *****          16956 17000 DAY=VAL(TEMP$(1,2)):PAYEE$=TEMP$(1
15050 ? DPNBR;" ";MONTH;"//";TEMP$(1,2 16503 P$(K,K)<">".) THEN NEXT K:RESP$(K)=".00 16958 17002 AMT=VAL(TEMP$(23,30)):MEMO$=TEMP$(1
15051 );TEMP$(7,14);";TEMP$(15,18)          16510 L=LEN(RESP$):FOR K=1 TO L:IF RES 16960 P$(31,48)
15052 ):TOT=TOT+VAL(TEMP$(7,14))          16511 P$(K,K)<">".) THEN NEXT K:RESP$(K)=".00 16962 17004 NUMSUB=VAL(TEMP$(49,49)):FOR J=1
15060 IF PFLAG=1 THEN LPRINT DPNBR;" 16512 IF L=K THEN RESP$(K+1,K+2)="00" 16964 TO NUMSUB:POS=40+10*j:SUBLEG(J)=VAL(TEM
15061 ";MONTH;"//";TEMP$(1,2);";TEMP$(7, 16513 IF L=K THEN RESP$(K+1,K+2)="00" 16966 P$(POS,POS+1))

```

CHECK CON'T

```

17040 SUBAMT(J)=VAL(TEMP$(P05+2,P05+9)
':NEXT J:RETURN
199 REM ***** CATEGORY #'S AND NA
THESE *****
20000 DATA "
20010 DATA Pas. Salary
20011 DATA Pas. Trans.
20012 DATA Rent/Ut/IRA
20013 DATA Pro. Growth
20014 DATA Leitch Sal.
20015 DATA AP Transp.
20016 DATA AP Health
20017 DATA AP Soc Sec.
20020 DATA Church Mort
20021 DATA Pkg Lot Ln.
20022 DATA Church Gas
20023 DATA Ch. Com Ed.
20024 DATA Ch. Water
20025 DATA Bond Int.
20026 DATA Bond Redem.
20027 DATA Bldg & Gnds
20030 DATA Off. Supply
20031 DATA Ch. Phone
20032 DATA Miscellany
20033 DATA Honorarium
20040 DATA Sunday Sch.
20041 DATA Youth
20042 DATA Bus Minist.
20043 DATA Awana Clubs
20044 DATA Jr. Church
20045 DATA VBS
20046 DATA Music
 47 DATA Dinners
 48 DATA Flowers
20060 DATA Rowells
20061 DATA Biven
20062 DATA Stirewalt
20063 DATA Brammer
20064 DATA Mingo
20065 DATA Buckley
20066 DATA Hayes
20067 DATA Miss. Other
20068 DATA F of Israel
20069 DATA Connon
20090 DATA B. Fund Dep
20091 DATA Others
20092 DATA Love Offer.
20093 DATA M. Fund Dep
20094 DATA Salables
20100 DATA ***** ADD DEPOSIT CATEGORI
ES IN THE 7 NEXT LINES *****
20101 DATA Love Off
20102 DATA Tithes
20103 DATA Missions
20104 DATA Bldg Fnd
20105 DATA Others
20106 DATA Awana
20107 DATA Extra

```

SMART CON'T

```

1510 ? " A...Z,";CHR$(34);" Marks (15),FILE2$(15):OFF=40000
cell as a label":? " 0...9,+,- Marks 105 GRAPHICS 0:POKE 82,1:POKE 83,38
cell as a value" 110 ? " Program Comparator":?
1515 ? " : Sum Formula. Total :? "This program analyzes two LISTed B
Is values":? " in a column ASICprograms and prints the difference
or row." S"
1520 ? " OPTION Marks cell as a f 130 ? :? :? "Please enter the name of
formula":? " SELECT Main Menu. Cho file #1"
ose from:-" 140 INPUT FILE1$
1525 ? " G=Global Format": 150 ? :? :? "Please enter the name of
? " L=Load Worksheet":? " file #2"
S=Save Worksheet" 160 INPUT FILE2$
1530 ? " P=Print Worksheet 170 ? :? :? " Please load disk con
";? " E=Erase Worksheet" taining both filesand hit RETURN to st
1532 ? " H=Home Cursor.(Ce art"
11 A1)":? " ?=Help Screen. 180 POKE 764,255
" 190 IF PEEK(764)=255 THEN 190
1535 ? " START Calculate Workshe 200 POKE 764,255
et":? :? :? "- PRESS ANY KEY TO RETUR 205 REM *** OPEN BOTH DISK FILES
N -";:GET #2,K:GOTO 420 206 CLOSE #1:CLOSE #2:CLOSE #3
1999 STOP 210 TRAP 220:OPEN #1,4,0,FILE1$:TRAP 0
2000 V$="1 2 3 4 5 6 7 8 910111213141 FF:GOTO 310
5161718192021222324252627282930313233 220 ? :? "FILE #1, (";FILE2$") PRODUC
4353637383940" ED AN ERROR, (";PEEK(195);")":?
2005 H$="ABCDEFGHIJKLMNO":H(1)=7:FOR T 330 TRAP OFF
=2 TO 4:H(T)=H(T-1)+8:NEXT T:POKE 5376 340 GOTO 130
0,200:GL=0 345 REM *** PROMPT FOR SCREEN VS PRINT
2010 OUT$(1)=" ";OUT$(4000)=" ";OUT$(2 ER
)=OUT$:IMS$=OUT$:SS$(1)=" ";SS$(600)=" "; 350 IO=0
SS$(2)=SS 360 ? :? "Do you want output routed to
2015 PR$(1)=" ";PR$(120)=" ";PR$(2)=PR printer?";
$ 365 POKE 764,255
2020 X=1:Y=1:C$="":XMIN=X:YMIN 370 IF PEEK(764)=255 THEN 370
=Y:Y1=X:X1=X:XMAX=4:YMAX=20:CELL$="A1" 375 IF PEEK(764)=43 THEN ? " Yes":IO=3
:FL=0:CL$="" "
2050 RETURN 380 IF PEEK(764)<>43 AND PEEK(764)<>35
2100 CLOSE #1:RESTORE 2120:POKE 54286, THEN GOTO 365
192 385 REM *** TRY TO OPEN THE PRINTER
2105 READ E,DI$:IF E=PEEK(195) THEN PO 390 TRAP 400:IF PEEK(764)=35 THEN ? "
SITION 8,1:? "Err:";DI$:TRAP 40000: No":? :GOTO 416
60SUB 210:GOTO 50 395 OPEN #3,8,0,"P":GOTO 416
2110 IF E=-1 THEN POSITION 8,1:? "Err 400 ? :? "PRINTER DOES NOT RESPOND":?
or:No.;"PEEK(195);":TRAP 4000 405 IF PEEK(764)=255 THEN 405
8:GOSUB 210:GOTO 50 NY KEY TO CONTINUE ":POKE 764,255
2115 GOTO 2105
2120 DATA 138,Device Not On,139,Device 410 CLOSE #3:GOTO 390
NAK,140,Serial Frame,142,Serial Bus,1 415 REM *** MAIN PROGRAM LOOP
43,Checksum No.143 416 POKE 764,255:IF IO=3 THEN PRINT #1
2122 DATA 144,Write Protected,162,Disk 0;"FILE #1 ";FILE1$:PRINT #IO;"FILE #2
Full,163,Bad DOS,167,File Locked,170, ";FILE2$:? #IO?:#IO
No such file,-1,a 420 LINE=0:TRAP 1000
425 INPUT #1:LINE1$
430 LINE1=VAL(LINE1$(1,5))
440 INPUT #2:LINE2$
450 LINE2=VAL(LINE2$(1,5))
460 IF LINE1>LINE2 THEN ? #IO?:#IO;"F
10 REM *** PROGRAM COMPARER BY RICK GR ILE #2 ONLY: ";LINE2$:GOTO 440
05ZKIEWICZ 470 IF LINE1=LINE2 THEN 500
100 DIM LINE1$(150),LINE2$(150),FILE1$ 480 ? #IO?:#IO;"FILE #1 ONLY: ";LINE1
$
```

FROM LAST MONTH
PROGRAM COMPARER

SMARTSHEET

SMARTSHEET is a spreadsheet calculator which is extremely handy for financial forecasting, budgeting or any calculations involving many variables. NOTE: Due to its need for two buffers, Smartsheet will only work on a 32K cassette or 48K disk system as a minimum. The on-screen worksheet is divided into cells or grid co-ordinates, arranged 15 columns across (A-O) and 40 rows down. Examples are: A1, O15, F32, etc. Due to the cell format (total of 600 cells) it is only possible to display a portion of the worksheet on the screen, so in order to view different areas of the worksheet, the screen acts as a scrolling window over the worksheet.

OPERATION: Those already familiar with spreadsheets (e.g. VISICALC, SYNCALC), should feel at home since Smartsheet is basically styled after VISICALC. To newcomers, I hope the following will be enough to get you started.

When Smartsheet is run, the screen is divided into two sections. The upper blue screen is the input window, which displays different menus, input prompts, error messages and the current cell co-ordinate. Below, is the grey worksheet screen, the window to the rest of the sheet. The black inverse bar is the cursor and is controlled by the normal cursor control keys. Its initial position is cell, A1. Smartsheet recognises three cell types: Labels, Values and Formulas.

Since Smartsheet only involves itself with number calculations, labels are for the users' benefit, similar to REM's in BASIC. They are usually placed in the column left of a value, to identify it. E.g., SALES, COST, PROFIT, etc. Labels are exactly like the list of items on a shopping list. To enter a label, position the cursor and type in the label, if the label is too long, the cursor will automatically be forwarded to the next column.

Values are numbers you input for the worksheet calculations to function properly. Values may take any form — positive, negative, decimal, etc. The use of values are similar to the prices next to items on a shopping list. Values are inputted by typing numbers directly into the cell. When the cursor is moved away, the value is moved to the right to align the decimal places. By pressing OPTION, formulas can be inputted into the current cell or answers to simple equations can be found.

NOTE: Values are necessary in a worksheet for formulas to function at all. Smartsheet gathers its input from cells nominated within a formula and displays the result after all calculations are complete. After pressing OPTION, 'Formula' appears on the status line and on the input line you are asked whether the first number in the formula is to be a cell location or a number.

The power of formulas in Smartsheet, is its ability to access values from other cells; i.e., formula may calculate a PROFIT figure, and therefore will access the values you inputted for SALES and COST and subtract them. Next the desired operation has to be inputted — addition, subtraction, multiplication, division or exponent (power of). Smartsheet is limited to one operation per formula. After entering the 2nd number as a cell or number the full equation will be seen on the sheet. If no cells have been accessed, the formulas will remain until they are calculated after pressing START.

A subset of the formula, is the SUM function, which is accessed by typing a colon (:). The SUM function allows you to total values between one cell and another in a particular row or column. After typing a colon, the input line asks "FROM CELL:?", here you should enter the cell where the totalling will begin, e.g. A1. Your input will be registered in the brackets in the status line. Input the cell, where the totalling will end, when "TO CELL:?" appears, e.g. A9. The input line will be cleared, and when you move the cursor off the formula cell, your From and To cells will be shown, e.g. :A1:A9.

Once you have finished structuring your worksheet, complete with labels, values and formulas, press START to calculate the worksheet. The message 'Calculating...' will appear while Smartsheet is computing answers. Calculating time depends on the number of formulas within the worksheet. When Smartsheet has finished, the screen will temporarily clear and the final worksheet will be seen with all formulas replaced with the results.

Smartsheet also has an optional menu for aid while developing a worksheet. The menu is accessed by pressing SELECT. The menu 'G L E S P H' will appear on the status line. Press the corresponding key to obtain these functions: Global Format: Selects how values are to be formatted when inputted. Choose from Dollar, Normal and Integer formatting. Dollar will automatically change your inputted value to dollar and cent format. Normal will leave your value untouched, while Integer will round your input to the next whole number.

Load Worksheet: Loads a previously saved worksheet from a disk or cassette. Press D or C to select Disk or Cassette respectively. If using cassette follow the same procedure as loading BASIC programs. If using disk, you may either press the bar to cycle through the Smartsheet workfiles on your disk and press RETURN to load the file displayed in the input window. Or you may input a filename directly on the input line and press RETURN to load it.

Save Worksheet: Saves current worksheet in memory to either disk or cassette. Press D or C to select Disk or Cassette to save on respectively. NOTE: Smartsheet saves the whole sheet, so cassette owners make sure you have about 50 counter spaces on the cassette and be prepared to wait during saving and loading times! Disk owners have two choices (same procedure as Load function): Press the space bar to cycle through the Smartsheet files on your disk and press RETURN to update or save over the file displayed in the input window or input a filename (8 letter limit) directly on the input line and press RETURN to save it. NOTE: Smartsheet uses 'SS' as an extender on its saved worksheets to identify them.

Erase Worksheet: Clears the current worksheet from memory. The program will re-ask whether you wish to erase the current worksheet in memory. Type Y to erase, or any other key to return to the worksheet. If you type Y then the screen will temporarily clear and a clean worksheet will appear.

Print Worksheet: Prints the current worksheet to a printer. Make sure your printer is ON LINE!! First you will be asked to input the cell at the lower right corner of your worksheet (in order to define the bottom and rightmost column). After entering the cell co-ordinate, you may imbed printer control codes at the beginning of each row in the worksheet. Type Y to imbed control codes e.g., double width for headings. If you elect not to use printer codes, press RETURN to begin printing. If you type Y for printer codes input the row number to imbed the code. NOTE: The program will send control code before printing the row. Next, type the code in and follow the same procedure to input more codes. When you have finished press RETURN to print the worksheet.

Home Cursor: Returns the cursor to cell A1. When you are moving around the far extremes of the worksheet it is handy to use this function instead of repeatedly using the cursor keys.

Help Screen: Calling up this screen lists all the main keys and functions of Smartsheet. That about wraps up the features of Smartsheet. If you prefer to have the cursor move without having to use the CONTROL and arrow keys simultaneously, just change the value equal to K in the lines 65, 70, 75 and 80 to 61, 45, 43 and 42 respectively.

EXAMPLE WORKSHEET: An actual example is better to explain the basics behind a worksheet or 'template', ala the two sample screens. Screen 1 shows a template in its raw state with all formulas being uncalculated. All headings and item names are example of labels. NOTE: Any character including numbers may be made into a label by typing an apostrophe before entering the label, e.g. the line of minus signs beneath the heading. The price of the items are all values and have been Dollar formatted. The formatting has been changed to 'Normal' mid-way in creation to prevent quantity values to be in dollar and cent format. In the D column, formulas are present. In cell D6, the value of cell B6 (price of chicken) will be multiplied by cell C6 (quantity of chicken). The result of this formula will be shown after calculation. The same applies to cell D18, where the item total (D15) is subtracted from the available cash (D3). Cell D15 (item total) is using the SUM function. Upon calculation, Smartsheet will add all values from cell D6 to D13. NOTE: Smartsheet calculates all formulas and sums from left to right, top to bottom on the worksheet. hen START is pressed, Smartsheet will pause to calculate and the result will appear as in Screen 2, where all formulas in the D column have been solved and replaced by a number. From here the user may experiment with different cash, price or quantity values to view the final outcome on Mrs Jones' purse. As can be seen in Screen 2, Mrs Jones will have trouble paying the bill with only thirty dollars.

```
490 INPUT #1;LINE1$:LINE1=VAL(LINE1$(1
,5)):GOTO 460
495 REM *** WE HAVE A MATCH!
500 IF LINE1$=LINE2$ THEN 425
510 ? #IO:? #IO;"FILE #1: ";LINE1$
520 ? #IO;"FILE #2: ";LINE2$
530 GOTO 420
1000 REM ***
1010 IF PEEK(195)=136 THEN ? :? "END O
F FILE REACHED":END
1020 ? "ERROR ";PEEK(195);" STOPPED EX
ECUTION "
```

STARWARS

STARWARS (Parker Brothers) is the translation of the famous Starwars arcade game put into cartridge for the Atari 8-bit computers. The object of the game is to destroy the Imperial Fighters, the maze of laser towers, and destroy the Death Star by flying down the trenches of the Death Star and dropping your cargo of anti-matter disrupter into the exhaust port. That's what I call what you are firing, for a lack of calling it anything else. They are actually Proton Torpedoes.

The game is played with the joystick as you might imagine, being an x-wing pilot, and the fire button is used to blast the Imperial Fighters, Towers, and Thermal Exhaust Port. I've noticed the controls are very similar to the arcade version!

Any time you want to freeze the action you only have to press (I didn't say Hit) the Space Bar. This allows you to take pictures of the screen to show your friends the ultimate high score.

There are 3 waves of difficulty which one has to deal with the destroy the Death Star:

1. Approaching the Imperial Fleet: You must destroy the Tie-Fighters. Sounds simple? Well, think again. These nasty little guys hurtle fireballs at you as they pass. The fireballs disrupt one shield per hit. You only have 9 energy shields to begin with! When successful you will be ported to the next wave, as with all the waves.

2. Surface of the Death Star: In this wave you maneuver your craft through a series of Laser Towers. The Towers shoot fireballs back at you, so be careful. You are to shoot the tops of the Towers to go on to the next wave. Sound simple? (tee hee) By the way, when you hit a tower with your ship you lose an energy shield.

3. Death Star Equatorial Trench (DSET): Once you are in the DSET be prepared to fly by the seat of your pants. Again as with the other waves there are the fireballs to put up with. You even get the chance to fly under or over suspended catwalks. Don't touch them with your craft or you will lose another shield. I've found it easier to blast the fireballs the instant they appear on the screen. Then all you have to do is fly between the catwalks.

When you have flown past the catwalks keep an eye on the trench floor for the Thermal Exhaust Port. Blast away to send your proton into the Death Star. Then off you go to watch the explosion, at which time you have saved the Rebel Base.

The game ends when you run out of shields then are hit by a fireball, run into a tower, or run into a catwalk. You get points along the way to save the Rebels by destroying the following: Tie Fighters — 1000 points; Fireballs — 33 points; Laser Towers — 200 point each (bonus of 50,000 points if all are destroyed); Death Star — 5000 points (bonus award of 3 energy shields).

Depending upon which wave you are in you will earn bonus points using the Force: Wave 1 — 5,000; Wave 2 — 10,000; Wave 3 — 15,000; Wave 4 — 25,000; Wave 5 — 50,000; Wave 6 — 100,000.

Upon playing the game I find myself wondering why I spent all those quarters in the arcade! The graphic ability of the Atari really shines in this program. My compliments to the game designer. The graphic screens in this game are very similar to those in the arcade. Well, very close for an 8-bit machine. There is also something worth mentioning. It has a 180 day warranty. Now that is something you don't see every day! I suggest this one be purchased and added to your collection.

— Stephen E. Warn

VP RAMBLINGS

The newsletter of late has not been the best that we can do. This has mainly been due to the fact that I have not coordinated my part of the putting together of the newsletter with the other editors. Anyway this policy is now changed and we will try to bring you the newsletter of old where everything in the issue was for that issue and nothing was left out. I hope that by these actions we will be able to bring you a better newsletter and one which all members can be proud of.

As you have seen in the stores and newsprint the price of computers is coming down and this means programs etc. are doing the same. From the rumors that are going around the price should fall even further so everyone who wants one can own a computer. This is good news for all of you who have been putting off buying some of that equipment you wanted. The way things are going you will soon be able to afford it. This should make things more enjoyable.

Keep your eyes on the printer market as they are not only coming down in price but they are getting better than ever. The quality of print is getting better, speed too. The noise level is lower along with the price so that, like computers, you can get great printers and very low prices especially compared to a few years ago.

More people of late have told me about buying a computer. They have talked about the ST and all that it can do and what they can do with it. After all that was said and done I asked them one question, "what do you want to do with the computer?" They thought about this and after they told me what they wanted it turned out all they need was a good old eight (8) bit machine. In fact the 130XE more than filled the bill and can do everything they wanted and more. So there is still a place for this machine and more software to operate it. The small machine is not dead, but very much alive and giving very useful service.

— Larry Gold

TOKENS

(Atari Tokenized BASIC Files, reprinted from October, 1985 R.A.G. BAG)

There is a pattern to the header seen on the first sector of a SAVED Atari BASIC file. The first 14 bytes (\$0D) are pointers to the various tables which BASIC uses to store variable names, values and the statements which make up our programs. Take a look at the sector dump below of the first sector of a BASIC program.

```
SECTOR 620 ($26C) FP. 621 F#13
00 00 00 01 2A 01 2B 01 ....*.+ Header
08 A3 01 B9 00 CF 0D 46 4E #.9.O.FN
10 A4 46 57 A4 D8 C2 4F 46 $FW$XBOF Var Name Table
18 46 53 45 D4 CB 43 54 52 FSETKCTR
20 CC D9 DA 44 53 44 4C 4C LYZDS DLL
28 CF 44 53 44 4C 48 C9 44 ODSLHID
30 53 44 CC 44 4C CC CC CA SDLDLLJ
38 00 80 00 00 00 00 00 ..... Var Value Table
40 00 80 01 00 00 00 00 00 .....
48 00 00 02 00 00 00 00 00 .....
50 00 80 03 00 00 00 00 00 .....
58 00 80 04 00 00 00 00 00 .....
60 00 00 05 00 00 00 00 00 .....
68 00 80 06 00 00 00 00 00 .....
70 00 80 07 00 00 00 00 00 .....
78 00 00 08 00 00 36 6D 7D ....6m.
```

The first two lines can be decoded by using the following diagram:
00 00 VN TB EV NT DV VT
DS TB DS DL DE ND vn vn

The first pair of bytes (\$00,01) is the pointer to LOMEM which is stored at \$80,81 (129,130 dec). It is loaded in from \$2E7,2E8 (743,744 dec). Again, these bytes are stored in lo-byte format, that is, the low half of the number is first and the high half is last. These first two bytes are always "00 00".

The second pair, labeled "VN TB" (bytes \$02,03) is the pointer (actually tells the distance of the Variable Name Table from the start of RAM) + 2567. These 256 bytes are a temporary storage buffer during tokenization of a statement just after you hit the RETURN key. The value is normally "00 01". I say normally because you can alter the value of VN, the third byte of the sector (& first byte of each additional header byte pair by the same amount), and effectively protect your program being listed. In our example "00 01" means the Variable Name Table starts at the \$0E (14 dec)th byte of the sector. The variable name is FN\$ and is "46 4E A4" in ATASCII. The variable names are written with the last character inverted. Dextral character inverted (DCI) means the high bit of the binary number used for the ending character is turned on making it inverse video. This in reality just adds \$80 (1000 0000 in binary), and in this way BASIC knows when it reaches the end of the variable name.

Next comes the pointer to the end of the Variable Name Table + 256 (bytes \$04,05 and shown as EV NT).

Bytes \$06,07 (shown as DV VT) gives the distance from the start of the variable name table to the start of the Variable Value Table + 256. Just count the number of bytes from the start of the variable name table to the start of the variable value table and add 256 (that's a hi-byte 01) and you've got the number.

In the second line of the sector dump, the bytes labeled "DS TB" (sector byte \$08,09) point to the start of the Statement Table. Following these bytes is the offset to the end of the Statement Table (bytes \$0A,0B shown as DS DL). This is the end of your BASIC program. These bytes point to the last byte of the last line of the actual program.

But there's more. Remember when you SAVED your program? The line you typed in with the program name is also there in your BASIC file. It's what I call the Default Save Last Line (DS DL). It starts at the byte following the end of the statement table (bytes \$0A,0B) and ends with the address pointed to by bytes \$0C,0D (labeled DE ND).

Look over the sector dumps included and see if you can find all the parts we've mentioned.

```

SECTOR 621 ($26D) FP. 622 F#13
00 00 00 00 00 00 09 00 00 ..... Var Value Table
08 00 00 00 00 00 0A 00 00 .....
10 00 00 00 00 00 0B 00 00 .....
18 00 00 00 00 00 0C 00 00 .....
20 00 00 00 00 00 0D 00 00 .....
28 00 00 00 00 00 0E 00 00 .....
30 00 00 00 00 0A 00 24 24 .... $$ Statement Table
38 00 41 54 41 52 49 20 42 .ATARI B
40 41 53 49 43 20 46 49 4C ASIC FIL
48 45 20 52 45 43 4F 56 45 3 RECOVE
50 52 59 2D 44 49 53 4B 9B RY-DISK.
58 14 00 13 13 00 62 79 20 .....by
60 42 69 6C 6C 20 50 65 74 Bill Pet
68 72 79 9B 1E 00 11 11 00 ry.....
70 31 37 20 44 65 63 20 31 17 Dec 1
78 39 38 34 9B 28 36 6E 7D 984.(6n.

```

The ending sector of the program is #646, with the last byte designating the number of data bytes in use. For us that's \$2B (43 dec).

SECTOR 646 (\$286) FP. O F#13

00 00 00 1B 1B 36 87 2D 036.-
08 41 02 08 00 00 00 16 E8 A.....h
10 03 06 06 24 16 00 80 16\$....
18 16 19 OF 0E 44 3A 52 45D:RE
20 43 4F 56 45 52 44 2E 42 COVERD.B
28 41 53 16 44 4C 48 C9 44 AS DLHD
30 53 44 CC 44 4C CC CA SDLDDLJJ
38 00 80 00 00 00 00 00
40 00 00 01 00 00 00 00
48 00 00 02 00 00 00 00
38 00 00 30 00 00 00 00
38 00 00 04 00 00 00 00
38 00 00 05 00 00 00 00
38 00 00 06 00 00 00 00
38 00 00 07 00 00 00 00
38 00 00 08 00 00 34 00 2B4.+

Tokens are an intermediate short-hand form of storing your BASIC statements. Each time you type in a line in BASIC and press RETURN, the line (presently in the line buffer) is tokenized and then stored in memory. When you RUN the program the tokens are quickly interpreted and translated to machine language and so goes the program. Two books are available: "De Re Atari" and "The Atari BASIC Sourcebook".

What follows is a list of tokens used in Atari BASIC (Atari Microsoft uses different token values):

Atari BASIC Fixed Tokens

Statement	Operator	Function
\$ #	\$ #	\$ #
00 0 REM	OE 14 (NUM CONST)	3D 61 STR\$
01 1 DATA	OF 15 (STR CONST)	3E 62 CHR\$
02 2 INPUT 10 16 DOUBLE QUOTE 3F 63 USR		
03 3 COLOR 11 17 (DUMMY)		40 64 ASC
04 4 LIST	12 18 ,	41 65 VAL
05 5 ENTER 13 19 \$		42 66 LEN
06 6 LET	14 20 :(STMT END)	43 67 ADR
07 7 IF	15 21 ;	44 68 ATN
08 8 FOR	16 22 CAR RTN	45 69 COS
09 9 NEXT	17 23 GOTO	46 70 PEEK
0A 10 GOTO	18 24 GOSUB	47 71 SIN
0B 11 GO TO 19 25 TO		48 72 RND
0C 12 GOSUB 1A 26 STEP		49 73 FRE
0D 13 TRAP	1B 27 THEN	4A 74 EXP
0E 14 BYE	1C 28 #	4B 75 LOG
0F 15 CONT	1D 29 (= (NUM)	4C 76 CLOG
10 16 COM	1E 30 ()	4D 77 SQRT
11 17 CLOSE 1F 31)=		4E 78 SGN
12 18 CLR	20 32 (4F 79 ABS
13 19 DEG	21 33)	50 80 INT
14 20 DIM	22 34 =	51 81 PADDLE
15 21 END	23 35 ±	53 82 STICK
16 22 NEW	24 36 *	54 83 PTRIG
17 23 OPEN	25 37 +	55 84 STRING
18 24 LOAD		26 38 -
19 25 SAVE		27 39 /
1A 26 STATUS 28 40 NOT		29 41 OR
1B 27 NOTE		
1C 28 POINT 2A 42 AND		
1D 29 XIO		2B 43 OR
1E 30 ON		2C 44)
IF 31 POKE		2D 45 = (ARITH)
20 32 PRINT 2E 46 = (STRING)		
21 33 RAD		2F 47 (= (STRINGS)
22 34 READ		30 48 (
23 35 RESTORE 31 49)=		
24 36 RETURN 32 50 (

25 37 RUN	33 51)
26 38 STOP	34 52 =
27 39 POP	35 53 + (UNARY PLUS)
28 40 ?	36 54 - (UNARY MINUS)
29 41 GET	37 55 ((STRING LEFT PAREN)
2A 42 PUT	38 56 ((ARRAY LEFT PAREN)
2B 43 GRAPHICS	39 57 ((DIM LEFT PAREN)
2C 44 PLOT	3A 58 ((FUNCTION LEFT PAREN)
2D 45 POSITION	3B 59 ((DIM STR LEFT PAREN)
2E 46 DOS	3C 60 , (ARRAY COMMA)
2F 47 DRAWTO	
30 48 SETCOLOR	
31 49 LOCATE	
32 50 SOUND	
33 51 LPRINT	
34 52 CSAVE	
35 53 CLOAD	
36 54 (IMPLIED LET)	
37 55 ERROR (SYNTAX)	

Each variable name you use in your program is also tokenized by a number starting with \$80+ (dec 128+) and referred to by the variable name (in sequence that you named them), rather than by their actual name. A short program to list out these tokens is given in the Atari BASIC sourcebook.

BASIC File Recovery

Protecting BASIC programs has been done for several years. In the Fall, 1983 issue of Atari Connection, p. 60, is given a simple way of protecting BASIC programs. I tried it and, indeed, it did work. In that article it is stated that once you protect your program it was irrecoverable. Irrecoverable is not impossible (at least not in the same way a formatted disk is). It took a couple of days. We'll use what I learned about the Default Save Last Line to demonstrate line structure and token use.

Resetting Current Line Pointer

32767 POKE

EKK(138)+256*PEEK(139)+2,0;SAVE"D:FILENAME":NEW
In tokens it looks like this:
FF 7F 45 33 01 46 3A 0E 41 01 38 00 00 00 2C 25 03 41 02 56 00
00 00 24 46 3A 0E 41 01 39 00 00 00 2C 25 0E 40 02 00 00 00 00
12 0E 00 00 00 00 00 00 14 42 19 0F 0A 44 3A 46 49 4C 45 4E 41
4D 45 14 45 16 16

This statement resets the line pointer for the current line. Recovery is by locating "DS DL" (bytes \$0A,0B) and finding the address of the start of the Default Save line (\$8000, 32767 dec). Using byte \$0E as the starting address, count forward (MSB-1)*256+LSB. This corrects for the buffer offset of the "DS DL" value. On reaching this, type in the SAVE line in accordance with the following example:

Default Last Line at End of BASIC program

Belajar Line Tokens

00 80 ____ 19 OF 08 44 3A - - - - - OE - - - 16
Line Byte Number

Bytes #2,3 are the line length and statement offset. If you use an eight letter filename plus a three letter extender you will end up with a line length of \$15 bytes. Enter your line length in both 2 and 3.

a line length of \$15 bytes. Enter your line length in both Z and S.

Once the default last line has been added at the proper location, add the line length (in hex) to the "DS DL" and place the new address (be sure it includes the MSB+1 buffer offset) in "DE ND" (\$C & D) of the BASIC program file header. This will enable you to list the BASIC program, providing the Variable Name Table is in order. If it has been replaced with single characters, like below, then edit by using normal CAPITAL letters (the hexadecimal ATASCII equivalence, of course), ended with an inverse capital letter or number. By the way, if you replace all the variable names with lowercase letters you will be able to list the program. However, when you attempt to list any line separately you'll end up with a mess. Give it a try and you'll see.

Replacing Variable Name Table with Single Character

32766 FOR Q=PEEK(130)+256*PEEK(131) T

Peeking prior to cleaning up the program will lock up the system, which precludes fixing the pointers and variable names that way. If all this seems like a lot of work, it is. However, I have summarized the cleaning-up process in a simple GET/PUT Basic program. This program, RECOVERD.BAS, has been on the RAG BBS (707-539-8889) for downloading.

A much better way to protect your BASIC programs, besides altering tables, is to compile them to machine code. This can be easily done with a BASIC compiler. MMG is a good one. What you end up with is a binary load file which can give you that "Assembly Language" programming ability without even the slightest knowledge of 6502 machine language.

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user group with no connection to Atari Corp. We are a group interested in educating members in the use of Atari computers and publishing the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world wide. Membership fees include the A.C.E. newsletter (10 issues per year) and high access in the A.C.E. Bulletin Board System. Dues are \$14 a year for U.S., and \$24 a year overseas airmail.

Subscription Dept.: 3662 Vine Maple Dr., Eugene, OR 97405
President — Dick Barkley, 2907 Wingate, Eugene, OR 97405

(503) 344-5843

Vice President — Larry Gold, 1927 McLean Blvd., Eugene, OR 97405
(503) 686-2490

8-Bit Librarian — Chuck & Jody Ross, 2222 Ironwood, Eugene 97401
(503) 343-5545

ST Librarian — Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405
(503) 484-4746

Editors — Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405
(503) 344-6193

Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405
(503) 484-4746

Larry Gold, 1927 McLean Blvd., Eugene, OR 97405
(503) 686-1490

E.R.A.C.E (Education SIG Editor) — Nora Young, 105 Hansen Lane
Eugene, OR 97404 / (503) 688-1458

Send 50c stamps or coin (\$1 overseas) to the Librarian for updated ACE Library lists.

A.C.E. Bulletin Board

(503) 343-4352

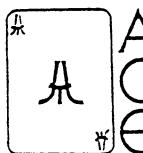
On line 24 hours a day, except for servicing and updating. Consists of a 130 XE upgraded to 576k by Charlie Andrews, 2 double-density ramdisks, 2 double-density disk drives, and a 10-meg hard disk from Supra Corp, an Epson MX80 printer, a 2400 US Robotics Courier modem, running the Mindlink Bulletin Board software (modified).

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct. 1981 to June, 1982.

Volume 2 contains July 1982 through June 1983.

Only \$12 each (\$2 extra for airmail). Available only from George Suetsugu, 45-602 Apuapu St., Kanoeha, HI 96744.



ATARI
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line". And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including the ACE Newsletter. Volume 1 covers April, 1981 through June, 1983. Volume 2 covers July, 1983 through December, 1985. Only \$6 per printed copy or \$11 per disk copy for ACE members. Available from: Jim Carr, 2660 S.W. DeArmond, Corvallis, OR 97333.

